

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE of
TECHNOLOGY

CARLOW

At the heart of South Leinster

Institute of Technology Carlow
Software Development

Final Report

Gym Performance System

Year 4 Project

By

Daniel Hayden - C00137009

Supervisor - Dr. Greg Doyle

Due Date: 18/04/2018

Table of Contents

1.0 Introduction	3
1.1 Source Control	3
1.2 Source Control	3
2.0 Description of Submitted Project	4
2.1 Web Application Framework	4
2.2 Web Application Overview	4
2.2.1 The Homepage (Logged Out)	5
2.2.2 Register	7
2.2.3 Login	8
2.2.4 Password Reset	9
2.2.5 Newsfeed	10
2.2.6 Client Manager (Logged in as Trainer)	12
2.2.7 View Requests (Logged in as Trainer)	13
2.2.8 Find User	14
2.2.9 Mailbox	15
2.2.10 View Mail	16
2.2.11 Create Mail	17
2.2.12 User Profiles	18
2.2.13 Newsfeed Edit Post	19
2.2.14 User Profile Settings	20
2.2.15 Exercise Plan Manager View (Logged in as Trainer)	21
2.2.16 Exercise Plan Manager Manage/Update/Edit (Logged in as Trainer)	22
2.2.17 Create Exercise Plan (Logged in as Trainer)	24
2.2.18 Trainer Finder (Logged in as Client/User)	27
2.2.19 My Exercise Plan (Logged in as Client/User)	28
2.2.20 View Exercise	29
2.3 Mobile Application Framework	30
2.4 Mobile Application Framework Overview	30
2.5 Mobile Application Screenshots	31
3.0 Description of Conformance to Specification and Design	34
4.0 Description of Learning	35
5.0 Review of Project	36

6.0 Acknowledgements

37

7.0 References

38

1.0 Introduction

The purpose of this document is to explain the work carried out when developing our gym training application called Total Fitness. We will also cover what we learned from this project and things that we would do differently if we were to start this project over.

As part of a fourth year of our Bachelor of Science in Software Engineering Honors we were required to develop a gym performance training application. As a requirement the application needed to run on multiple devices and it needed to be secure. We feel that our final two products (Web and Mobile based applications) meet these requirements as set out.

1.1 Code quality and security

When developing our code we had a strong focus on code quality and security. The laravel framework provided us with the security that we needed and allowed us to write clean high quality code. High security was one of the features mentioned in a original specification and laravel allowed us to do that by creating secure user authentication, password encrypting, cross site request forgery prevention, cross site scripting prevention and much more.

1.2 Source Control

This entire project, Web application and mobile, was created using source control manager git and is stored on github at:

Web Application

<https://github.com/Sledro/TotalFitnessMobile>

Mobile

<https://github.com/Sledro/TotalFitnessLaravel>

2.0 Description of Submitted Project

Please note that all exercises are pre created and inserted into the databases exercise table manually. The personal trainer are then able to select from these predefined exercises when creating their plan. This was done to ensure that all of the exercises will contain the correct information and rain uniform.

2.1 Web Application Framework

The web application was created using the PHP framework Laravel - “The PHP Framework For Web Artisans” [1]. This was our first time using laravel and overall it was a great experience. Laravel is installed using a PHP package and dependency manager called “Composer” [2]. Laravel is a MVC (Model View Controller) framework that provide the quick and efficient creation of models views and controllers via the command line using its command line interface called Artisan. Artisan is the command-line interface included with Laravel. “It provides a number of helpful commands that can assist you while you build your application” [3]. Laravel also provides you with the option to use database migrations. Database migrations are essentially source control for your database while it is under development. For a project, we did use database migrations for a few weeks and then decided against it as it was greatly slowing down development time trying to learn how to properly use database migrations, PHP MVC and the Laravel Framework itself. Another useful tool that Laravel provides you as a developer is the ability to create easy object or model relationships using Eloquent. “The Eloquent ORM included with Laravel provides a beautiful, simple ActiveRecord implementation for working with your database. Each database table has a corresponding "Model" which is used to interact with that table. Models allow you to query for data in your tables, as well as insert new records into the table.” [4]. There is a many more features and advantages to name here that using a PHP Framework such as Laravel provided us. The above features are an overview of the main features that we used while developing Total Fitness.

2.2 Web Application Overview

In this section we will present screenshots of the multiple different pages that are on the web application, the purpose and and the functionality and work carried out for that page. Please note that instead of going through all of the code here that we wrote, we will focus on key functions. All of the code developed by us for this project along with instructions on how to install both applications can be found in our Technical Manual. Please note that all of the following pages use web routes to define which url when accessed should point to which controller and return which view. Agin all of the routes are posts in the Technical Manual.

2.2.1 The Homepage (Logged Out)

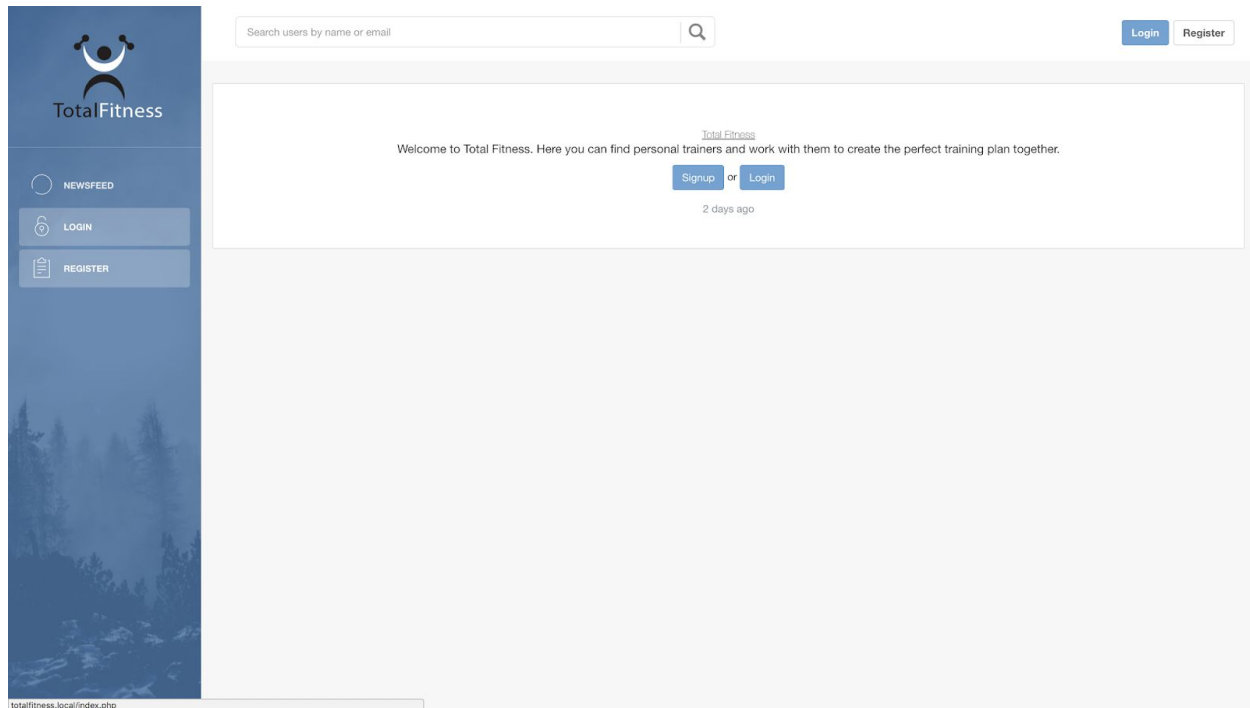


Figure.1 The homepage when no user is logged in

Purpose

This is the landing page for when a user who is not logged in or does not yet have an account accesses the web application. The user will be presented with only three options in the menu bar on the left. This is because we do not want a user accessing pages that require an authenticated session to function correctly. It also encourages the user to sign up to a website. The user is given the options to sign up, login or view the newsfeed. When a user clicks on the newsfeed they will be asked to register before they can view it.

Functionality and summary of work carried out

To set up this page we first needed to set a route

```
Route::get('/', 'PagesContorller@index');
```

When a user views this url the index function inside the PagesController is called.

```
public function index(){  
    if(Auth::guest()){  
        return view('pages.index');  
    }  
    else{  
        return redirect ('/newsfeed');  
    }  
}
```

This function then checks if the user visiting the index page is authenticated or not. If the user is not a guest then they will be redirect to the newsfeed.

2.2.2 Register

Search users by name or email

Sign up to Total Fitness for free!

USERNAME
Sledrd

NAME
Daniel Hayden

E-MAIL ADDRESS
dan@sledro.com

PASSWORD
.....

CONFIRM PASSWORD
.....

ARE YOU A PERSONAL TRAINER?
No

Register

Figure.2 The register page

Purpose

The purpose of this page is to allow the user to register or create an account.

Functionality and summary of work carried out

This page was generated using Laravel's CLI command called "php artisan make:auth". [5] This command tells Laravel that we want to use user authentication and Laravel will generate the registration, and login page for you securely. By doing this Laravel will also look after secure user sessions for you. This page was slightly modified by ourselves to take an extra 'dropdown select' parameter called isTrainer. This variable stores in our Users table in a database informing us whether or not a user is a personal trainer.

```
'isTrainer' => $data['isPT']
```

Passwords are securely encrypted before by Laravel they are stored in the database.

```
'password' => bcrypt($data['password']),
```


2.2.3 Login

The image shows a web application interface for 'TotalFitness'. On the left is a dark blue sidebar with the logo and navigation options: NEWSFEED, LOGIN, and REGISTER. The main content area is white and contains a search bar at the top with the text 'Search users by name or email' and a magnifying glass icon. Below the search bar is a login form. The form has two input fields: 'E-MAIL ADDRESS' with the value 'dan@sledro.com' and 'PASSWORD' with masked characters. There is a 'REMEMBER ME' checkbox and a 'Login' button. A link for 'Forgot Your Password?' is also visible. In the top right corner of the main content area, there are 'Login' and 'Register' buttons.

Figure.3 The login page

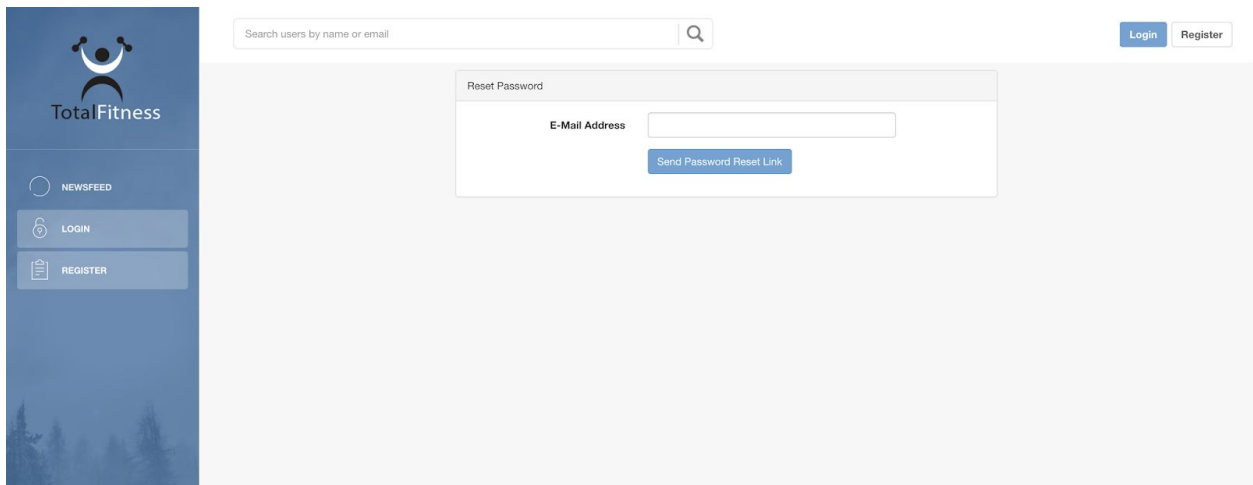
Purpose

To allow a registered use to login and create a secure user session with the web application.

Functionality and summary of work carried out

As with the registration page above, this page was created using “php artisan make:auth”. [5]

2.2.4 Password Reset



The screenshot shows the TotalFitness website interface. On the left is a dark blue sidebar with the TotalFitness logo at the top, followed by navigation links: NEWSFEED, LOGIN, and REGISTER. At the top right of the main content area, there is a search bar labeled "Search users by name or email" and buttons for "Login" and "Register". The central focus is a "Reset Password" form. This form contains a single input field labeled "E-Mail Address" and a blue button labeled "Send Password Reset Link".

Figure.4 The password reset page

Purpose

The purpose of this page is to provide users with the ability to reset their password via their email address in the event that they lost their password.

Functionality and summary of work carried out

As with the registration page above, this page was created using “php artisan make:auth”. [5]

2.2.5 Newsfeed

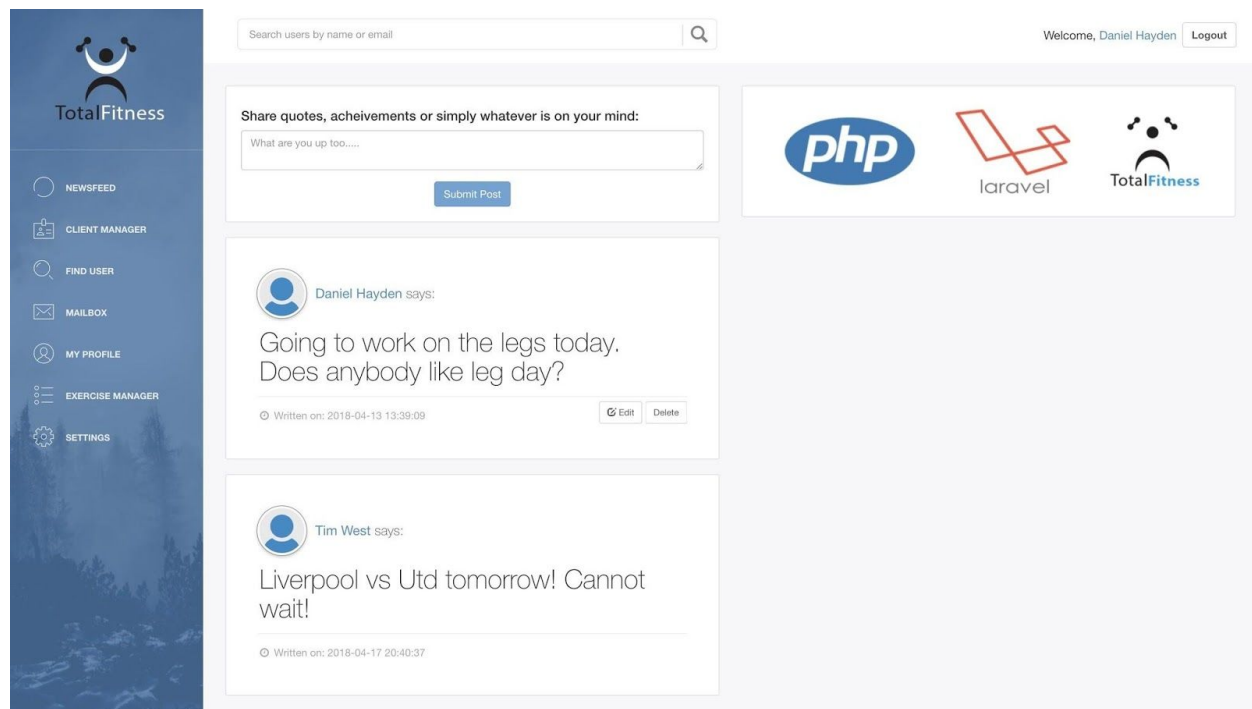


Figure.5 The Newsfeed

Purpose

We knew we wanted to make a site more interactive and we thought what better way to achieve this then by adding a news feed that users can post messages to. We added a sponsors section on the right hand column of this web page, to remove blank space on the web application and allow for more revenue. The idea here is that gym training supplements and gyms themselves can rent advertising space here. This column will not be displayed on mobile.

Functionality and summary of work carried out

This page consists of three main files. The Newsfeed model, NewsfeedController and the Newsfeed->index.blade.php view. This page uses the table in the database called newsfeed, which stores the user's posts to the newsfeed.

In a model we defined a one to many relationship between the newsfeed and the user model.

```
protected $table = 'newsfeed';

public function users() {
    return $this->belongsToMany('App\User', 'userID', 'id');
}
```

In the controller we have the functions that are used for carrying out the multiple different tasks required for this page to function. Index() will fetch all posts from the database and return them to this view as an object to be displayed. Here we also have similar functions for storing(), editing() updating() and deleting() posts on the newsfeed.

The SQL queries in this class are written to only allow the user to see the any posts that are created by a user that they follow or that are created by themselves. Here is an example of the index()

```
$userID=Auth::user()->id;
$userID2=Auth::user()->id;
$posts = Newsfeed::select('*')
->from('newsfeed')
->whereRaw('userid IN ( SELECT userID FROM user_followers WHERE
followerID=:userID ) OR userID=:userID2' )
->take(10)
->setBindings([$userID, $userID2])
->get();
return view('newsfeed.index')->with('posts', $posts);
```

Lastly in this controller we have a function called profileFeed(\$id). This function is used for retrieving all the posts on the newsfeed that were made by the user with the given \$id which is what we need for a users profile page newsfeed.

2.2.6 Client Manager (Logged in as Trainer)

The screenshot shows the TotalFitness Client Manager interface. On the left is a dark blue sidebar with the TotalFitness logo and navigation menu items: NEWSFEED, CLIENT MANAGER (highlighted), FIND USER, MAILBOX, MY PROFILE, EXERCISE MANAGER, and SETTINGS. The main content area has a search bar at the top with the text 'Search users by name or email' and a magnifying glass icon. To the right of the search bar, it says 'Welcome, Daniel Hayden' and a 'Logout' button. Below this is the 'Client Manager' title and a green 'View Requests' button. A table lists three clients:

CLIENT ID	NAME	USERNAME	EMAIL	PLAN	UPDATE	REMOVE
1	Daniel Hayden	Sledro	dan@sledro.com	Advanced Plan ▾	Update Plan	Remove as Client
2	Jim Smith	Test	jim@smith.com	test ▾	Update Plan	Remove as Client
3	Paul West	Tester	paul@west.com	Advanced Plan ▾	Update Plan	Remove as Client

Figure.6 Client Manager

Purpose

The purpose of the client manager is to allow personal trainers to view the entire list of their existing clients and to allow the personal trainer to manage or update the training plan that the trainer currently has assigned to each client. The personal trainer also has the ability to remove a client on this page. Finally the personal trainer has a green button on this page to allow them to view any requests that they might have received from users to become their personal trainer.

Functionality and summary of work carried out

The main files used to create and manage this page are the ClientManager model, ClientManagerController and the ClientManager>viewClients.blade.php view. This page uses the column users table in the database called trainerID. This trainerID column stores the ID of the current trainer assigned to that user.

2.2.7 View Requests (Logged in as Trainer)

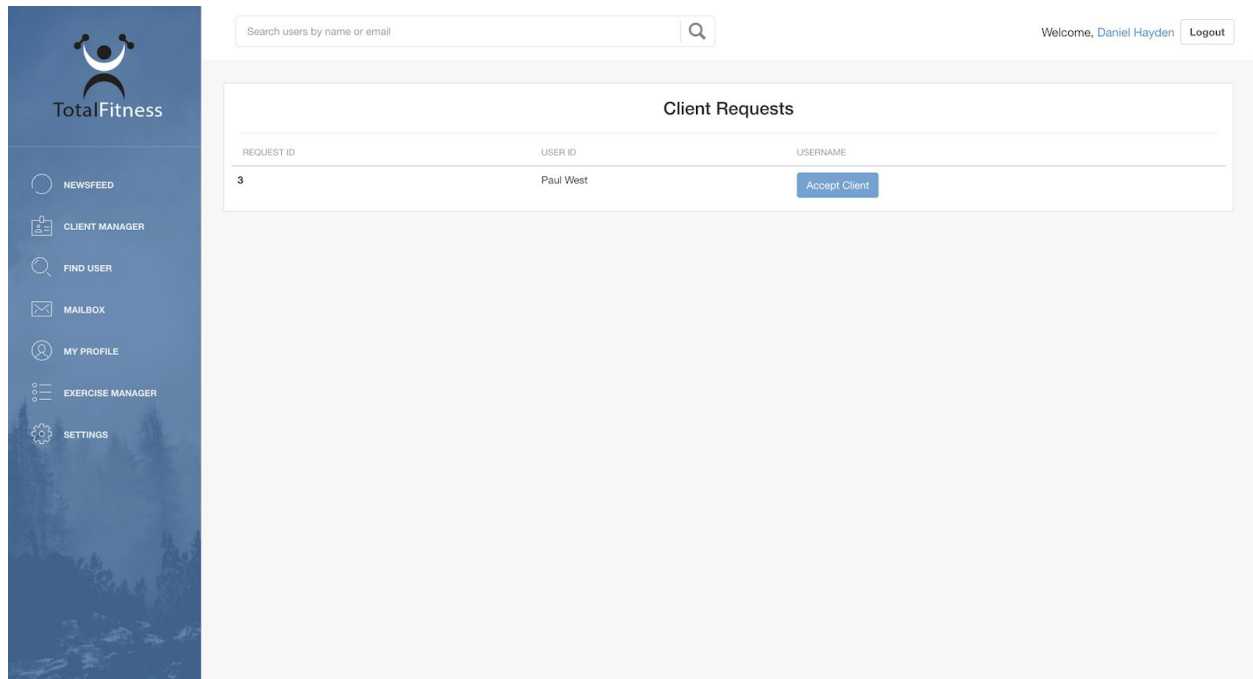


Figure.7 View Requests

Purpose

This page was created to allow a personal trainer to view and accept (or ignore) all of the current users that have requested this personal trainer to become their own personal trainer.

Functionality and summary of work carried out

The main files used to create and manage this page are the ClientManager model, ClientManagerController and the ClientManager>requests.blade.php view. The controller have two functions called requests() and accept(). These functions contain the logic for how client requests are handled. We also added a trainer_requests table into the database for this page. This tables give us the ability to store a record of every request made.

2.2.8 Find User

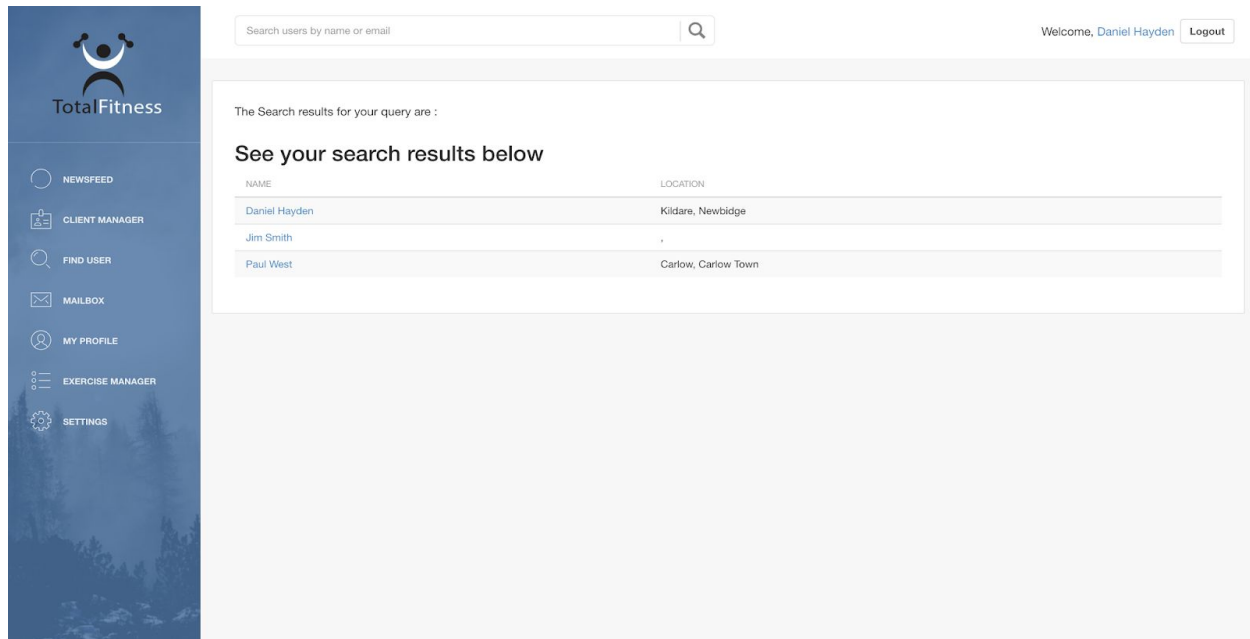


Figure.8 Find user page

Purpose

This page was created to allow any user to search for another user or personal trainer by name or email.

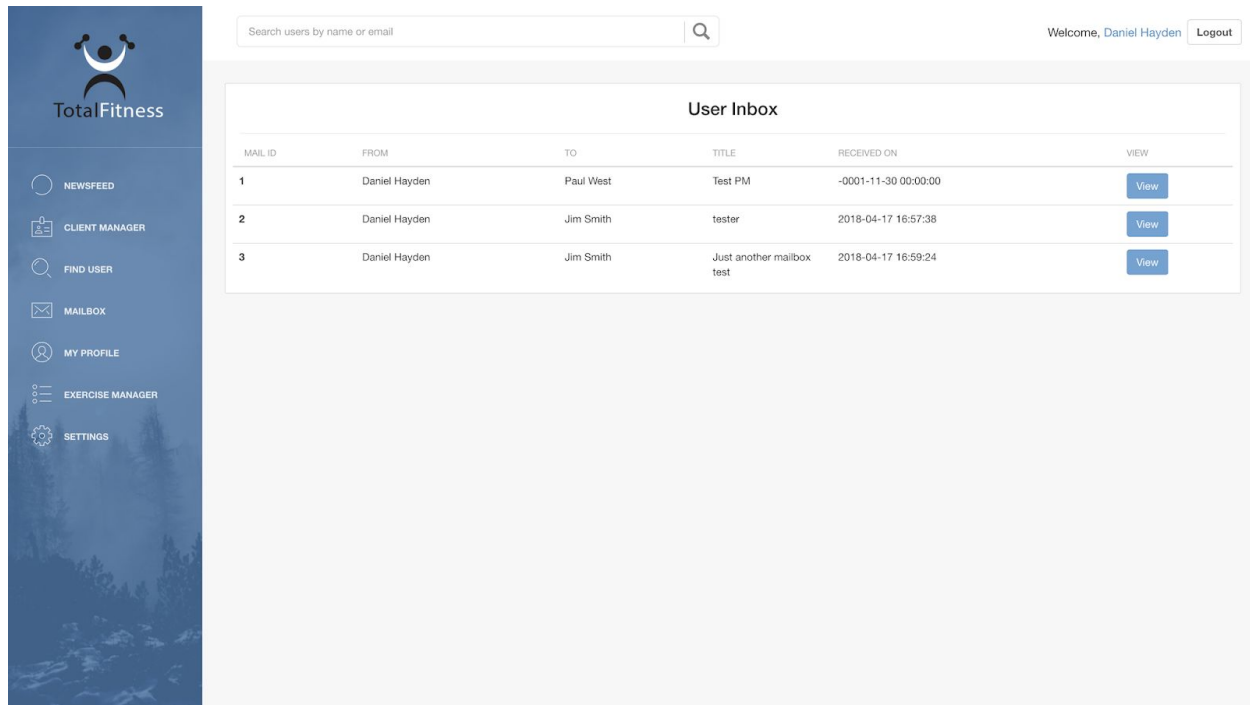
Functionality and summary of work carried out

The main files used to create and manage this page are the routes file located at routes->web.php and the search.blade.php view. The search menu in the top menu bar is also included at inc-> navbar.blade.php. This page gives a users the ability to search for another user or personal trainers profile using their full name or part of their name. A user can also be found by searching for their email address. Most of the logic for this page could be coded directly in the routes page

```
Route::any('/search',function(){
    $q = Input::get ( 'q' );

    $user = User::where('name','LIKE','%'.$q.'%')->orWhere('email','LIKE','%'.$q.'%')->get();
    if(count($user) > 0)
        return view('search')->withDetails($user)->withQuery ( $q );
    else return view ('search')->withMessage('No Details found. Try to search again
!');});
```

2.2.9 Mailbox



The screenshot shows the 'User Inbox' page in the TotalFitness application. The sidebar on the left contains navigation links: NEWSFEED, CLIENT MANAGER, FIND USER, MAILBOX, MY PROFILE, EXERCISE MANAGER, and SETTINGS. The main content area features a search bar at the top with the text 'Search users by name or email' and a magnifying glass icon. To the right of the search bar, it says 'Welcome, Daniel Hayden' and a 'Logout' button. Below this is a table titled 'User Inbox' with the following data:

MAIL ID	FROM	TO	TITLE	RECEIVED ON	VIEW
1	Daniel Hayden	Paul West	Test PM	-0001-11-30 00:00:00	View
2	Daniel Hayden	Jim Smith	tester	2018-04-17 16:57:38	View
3	Daniel Hayden	Jim Smith	Just another mailbox test	2018-04-17 16:59:24	View

Figure.9 View Mailbox

Purpose

The purpose of this page, is to allow users to view a record of all of the private messages that they have either sent or received.

Functionality and summary of work carried out

The main files used to create and manage this page are the Mailbox model, MailboxController and the Mailbox>viewAll.blade.php view. The controller has a function for fetching and returning all of the user's private mails called index() that returns the mails object to the view.

```
public function index()
{
    $mail = Mailbox::with('sender')->with('receiver')->where('receiverID', '=', Auth::user()->id)->orWhere('senderID', '=', Auth::user()->id)->get();
    return view('mailbox.viewAll')->with('mail', $mail);
}
```


2.2.10 View Mail

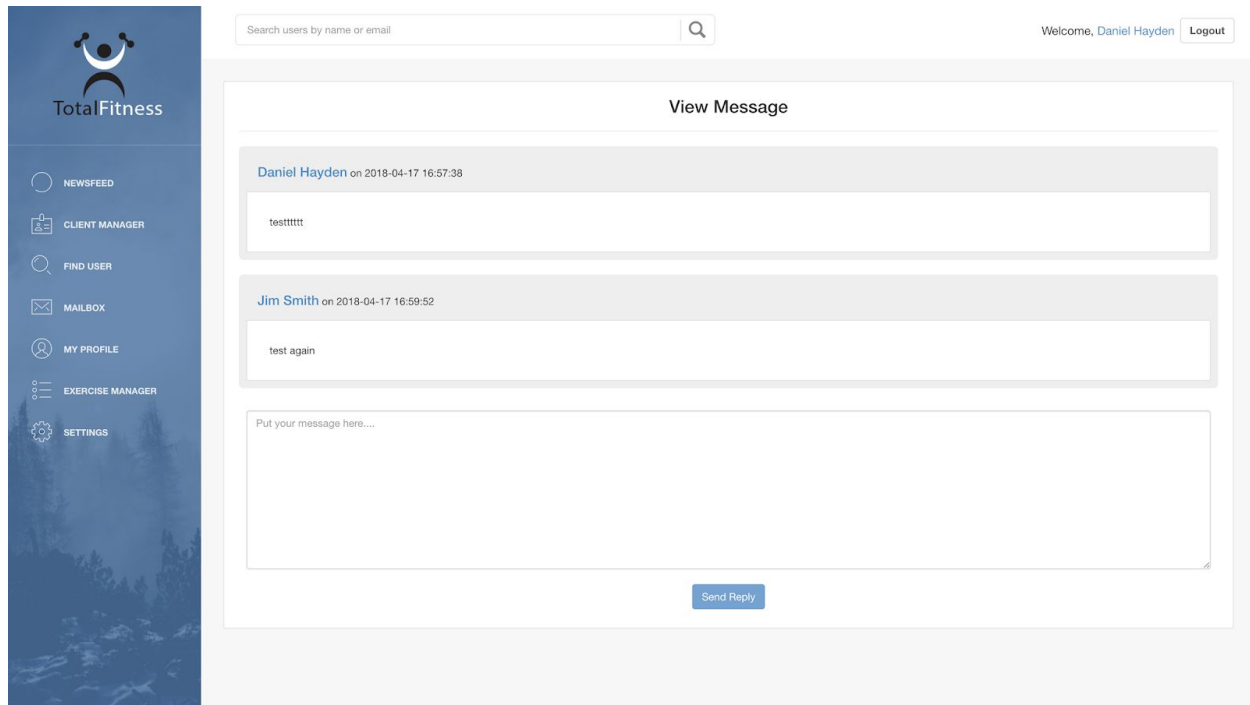


Figure.10 View Mail

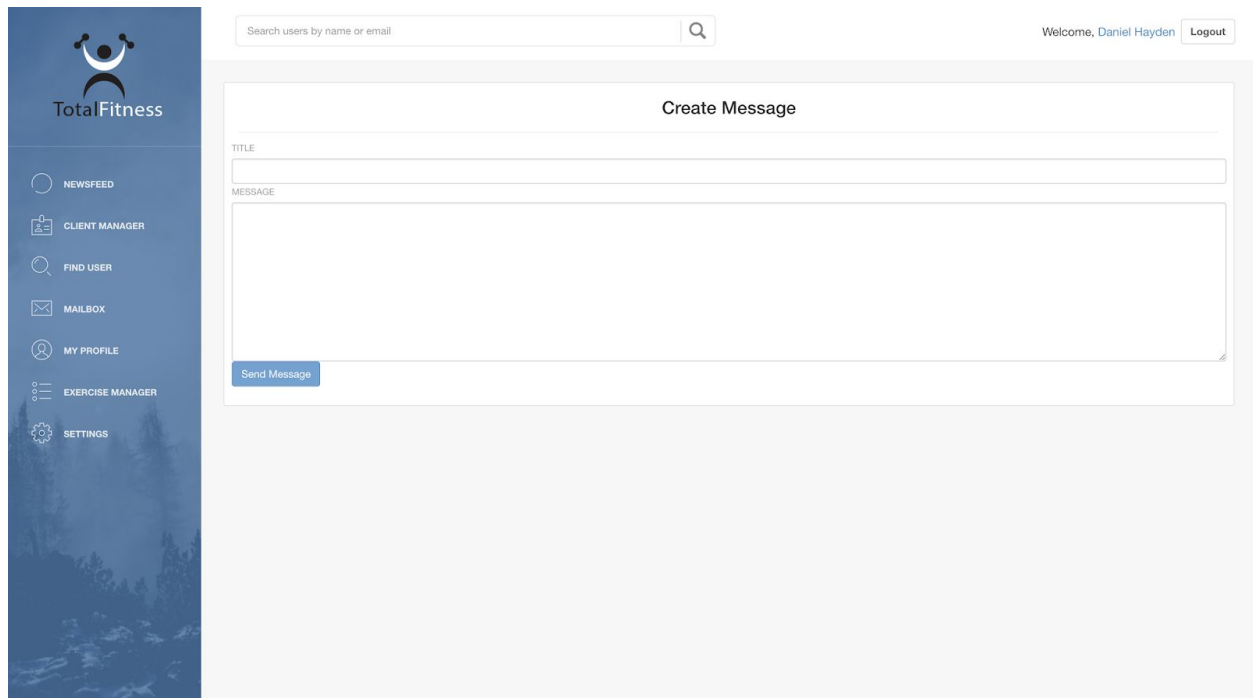
Purpose

The purpose of this page, is to allow users to view an individual mail and to allow the user to reply to that mail.

Functionality and summary of work carried out

The main files used to create and manage this page are the Mailbox model, MailboxController and the Mailbox>viewMessage.blade.php view. The controller has a function for fetching and returning all of the messages that are included in the given mail. The controller also provides a function to submit the reply which is called reply(). The reply comes from a form that is coded inside the view.

2.2.11 Create Mail



The screenshot shows the 'Create Message' form in the TotalFitness application. The form is titled 'Create Message' and contains two input fields: 'TITLE' and 'MESSAGE'. Below the 'MESSAGE' field is a 'Send Message' button. The form is located in the main content area of the application. On the left side, there is a dark blue sidebar with the TotalFitness logo and a list of navigation items: NEWSFEED, CLIENT MANAGER, FIND USER, MAILBOX, MY PROFILE, EXERCISE MANAGER, and SETTINGS. At the top of the page, there is a search bar with the text 'Search users by name or email' and a magnifying glass icon. In the top right corner, there is a user greeting 'Welcome, Daniel Hayden' and a 'Logout' button.

Figure.11 Create Mail

Purpose

This page allows a user to create a new private mailbox message and send it to a user. This page can be accessed by clicking on the 'Private Message' button located on every user's profile page.

Functionality and summary of work carried out

The main files used to create and manage this page are the Mailbox model, MailboxController and the Mailbox>create.blade.php view. The view contains a form that is submitted to the controller via the web routing page. The functions called store() which is located inside the controller will then take the values from the form and store them in a database. There are two tables in the database that are used to store user mail. These are mailbox and mailbox_messages.

2.2.12 User Profiles

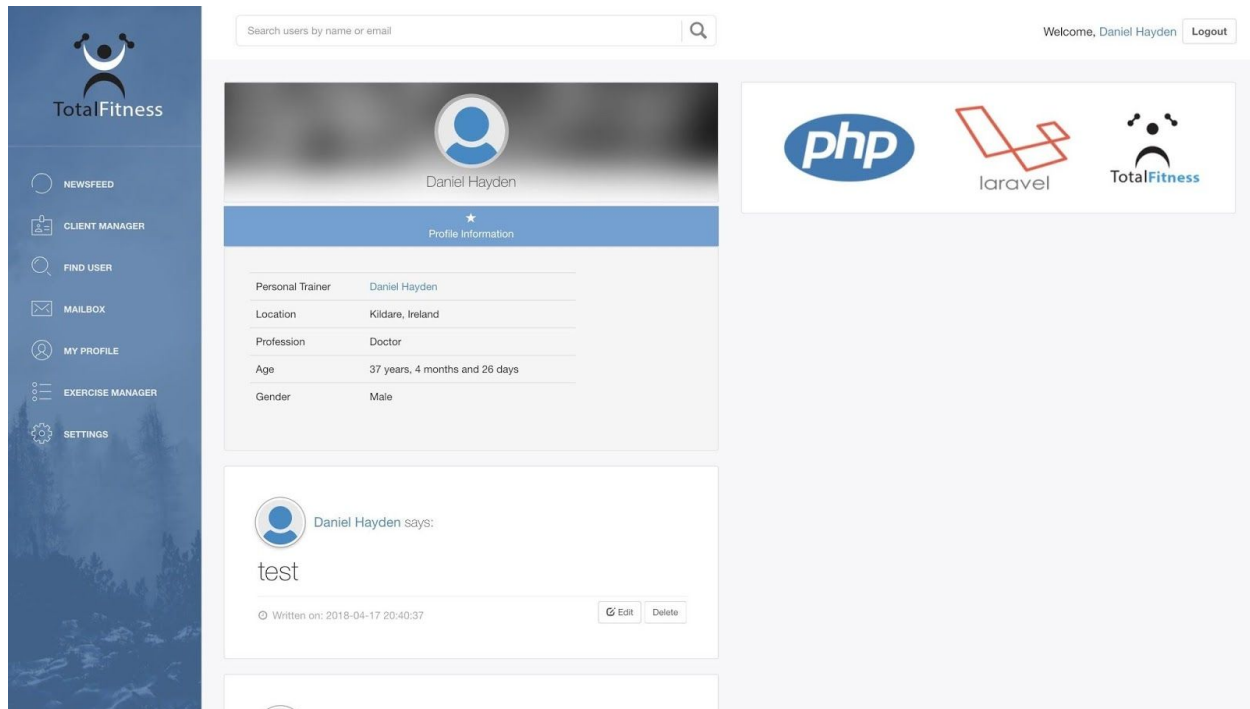


Figure.12 User Profiles

Purpose

The user profile page is used to display details about the user such as who their personal trainer is, where they are located and their age. This page also shows any posts that the user has made to the newsfeed and a sponsors section.

Functionality and summary of work carried out

The main files used to create and manage this page are the User model, UserController and the User>index.blade.php view. The username is taken from the URL via a http GET request in the route. The route then sends that username to the controller called show(\$username) which will then return the user details to the view. Users are stored in the database table called users.

2.2.13 Newsfeed Edit Post

The screenshot shows the 'Edit Post' interface. On the left is a dark blue sidebar with the 'TotalFitness' logo and a list of menu items: NEWSFEED, CLIENT MANAGER, FIND USER, MAILBOX, MY PROFILE, EXERCISE MANAGER, and SETTINGS. The main content area has a light gray background. At the top right, there is a search bar with the placeholder text 'Search users by name or email' and a magnifying glass icon. Next to it is a user greeting 'Welcome, Daniel Hayden' and a 'Logout' button. The 'Edit Post' title is centered at the top of the main area. Below the title is a form with two main sections: 'Title' and 'Body'. The 'Title' section has a text input field containing the word 'title'. The 'Body' section has a larger text area containing the word 'test'. Below the text area is a blue 'Submit' button.

Figure.13 Newsfeed edit post page

Purpose

This page allows a user to edit a post that they have created on the newsfeed. Note: The user also has the option to delete a post directly on the newsfeed.

Functionality and summary of work carried out

This page consists of three main files. The Newsfeed model, NewsfeedController and the Newsfeed->edit.blade.php view. The form is submitted to the controller along with the postid and the controller will validate the input data and update the corresponding record in the database.

2.2.14 User Profile Settings

The screenshot shows the 'Profile Settings' page for a user named Daniel Hayden. The page has a dark blue sidebar on the left with the 'TotalFitness' logo and navigation menu items: NEWSFEED, CLIENT MANAGER, FIND USER, MAILBOX, MY PROFILE, EXERCISE MANAGER, and SETTINGS. The main content area is white and contains a search bar at the top with the text 'Search users by name or email' and a magnifying glass icon. To the right of the search bar, it says 'Welcome, Daniel Hayden' and a 'Logout' button. The 'Profile Settings' form includes the following fields:

- EMAIL ADDRESS: dan@sledro.com
- FULL NAME: Daniel Hayden
- STREET: 126 Moorefield Park,
- TOWN: Newbridge
- COUNTY: Kildare
- COUNTRY: Ireland
- DATE OF BIRTH DD/MM/YY: 23-11-1980
- PROFESSION: Doctor
- GENDER: Male

At the bottom of the form is a blue 'Update Profile' button.

Figure.14 User settings page

Purpose

The purpose of this page is to allow the user to add or update their profile information such as Name, Address or Gender.

Functionality and summary of work carried out

The main files used to create and manage this page are the User model, UserController and the User>insettingsdex.blade.php view. The form is submitted to the controller along with the user id and the controller will validate the input data and update the corresponding record in the database.

2.2.15 Exercise Plan Manager View (Logged in as Trainer)

The screenshot displays the 'Update Training Plan' interface. At the top left is the TotalFitness logo. A search bar for users is at the top center, and a welcome message for Daniel Hayden with a Logout button is at the top right. The main content area features a table with the following data:

PLAN ID	NAME	# ACTIVE CLIENTS	CREATED ON	MANAGE	DELETE
29	test	55	2018-04-17 09:32:53	Manage	Delete (Cannot be undone)
30	Advanced Plan	55	2018-04-17 10:39:39	Manage	Delete (Cannot be undone)

A green 'New Exercise Plan' button is positioned in the top right corner of the table area.

Figure.15 Exercise Plan Manager page

Purpose

This is one of the main pages in the application. It allows the logged in personal trainer to view all of their pre-created training plans. The personal trainer can manage or update an individual training plan by clicking on the blue 'Manage' button for the selected plan. The personal trainer can also navigate to the create a new exercise plan page by click on the green button on the top right hand side of this page. Finally this page provides the personal trainer with the ability to delete the selected training plan by click on the red 'Delete' button located on each plan/row.

Functionality and summary of work carried out

The main files used to create and manage this page are the ExercisePlans model, ExercisePlansController and the ExercisePlans>viewPlansTrainer.blade.php view. The authenticated trainer id is sent from the session to the controller which returns all of the training plans that are associated with each use to the view, The object of that is return contains all of the plans and is iterated through inside the view to populate the table.

2.2.16 Exercise Plan Manager Manage/Update/Edit (Logged in as Trainer)

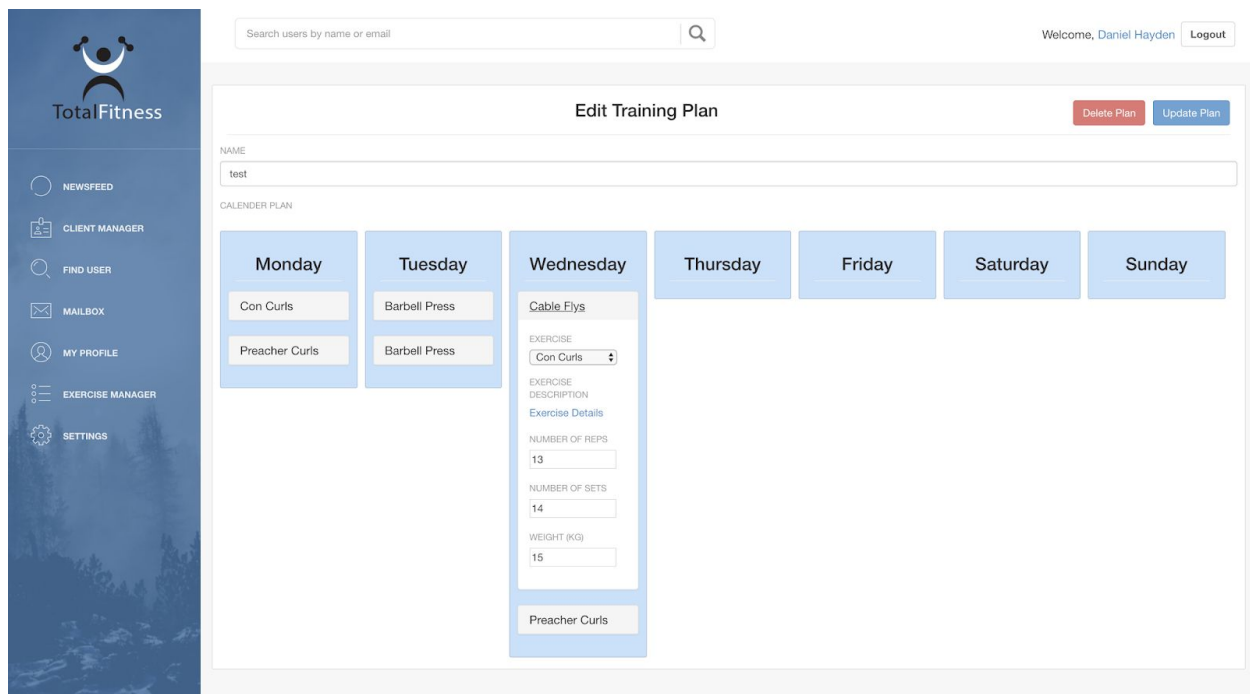


Figure.16 Exercise Manager page

Purpose

The purpose of this page is to allow the personal trainer to change or update or delete a training plan that they have already created.

Functionality and summary of work carried out

The main files used to create and manage this page are the ExercisePlans model, ExercisePlansController and the ExercisePlans>editPlan.blade.php view. The plan id is received from the URL via a http GET request. This id is send to to the controllers show() function and this function returns the exercise plan details along with the exercises included in that plan from the database tables: exercise_plan, exercise_plan_details and exercises. This can be done in one call using Laravel Eloquent Object Relation Model.

```
$plan = ExercisePlan::with('exercises.exercise')->find($id);  
$exerciseList = Exercise::pluck('name', 'id');  
  
return view('exercisePlans.editPlan')->with('plan', $plan)->with('exerciseList',  
$exerciseList);;
```

These relationships are set up in the corresponding models

//Exercise Plan Model has many relationship for ORM

```
public function exercises()  
{  
    return $this->hasMany('App\ExercisePlanDetails', 'planID', 'id');  
}
```

//Exercise Plan Details Model

```
public function exercise()  
{  
    return $this->belongsTo('App\ExercisePlanDetails', 'exerciseID');  
}
```

Each day of the week is collapsible to make the page shorted on a mobile display. This also works quite well with the desktop display so we decided to keep that there also. When the exercise details is clicked on it will navigate the user to view that exercises on the corresponding exercise page.

One this particular plan, Thursday to Sunday are rest days are no exercise have been submitted for those days.

2.2.17 Create Exercise Plan (Logged in as Trainer)

The screenshot displays the 'Create Training Plan' interface. At the top left is the TotalFitness logo. A search bar is located at the top center, and a user greeting 'Welcome, Daniel Hayden' with a 'Logout' link is at the top right. The sidebar on the left contains navigation links: NEWSFEED, CLIENT MANAGER, FIND USER, MAILBOX, MY PROFILE, EXERCISE MANAGER, and SETTINGS. The main content area is titled 'Create Training Plan' and includes a green 'Create this Plan' button. Below the title is a form for naming the plan, with the text 'Exercise Plan Name E.g Advanced Bulking'. The main section is a 'CALENDER PLAN' with seven columns for the days of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. Each day's column has an 'Add Exercise' button and a 'New Exercise' form. Each form contains an 'EXERCISE' dropdown menu (currently showing 'Con Curls'), an 'EXERCISE DESCRIPTION' field, and input fields for 'NUMBER OF REPS', 'NUMBER OF SETS', and 'WEIGHT (KG)'.

Figure.17 Create exercise plan

Purpose

The purpose of this page is to allow the personal trainer to create a new exercise plan.

Functionality and summary of work carried out

The main files used to create and manage this page are the ExercisePlans model, ExercisePlansController and the ExercisePlans>createPlan.blade.php view. We created an algorithm inside the view to add new divs to the view with unique div id numbers. This is done using javascript. This page was the first page we developed using the weekly training plan calendar design and it took quite a lot of time to get right. Again everything needed to be mobile responsive.

Add new div via javascript algorithm

```
<script>
var value = "";
function changeVal(choice) {
  if (choice == 'val1') {
    value = '1';
  } else if (choice == 'val2'){
    value ='2';
  } else if (choice == 'val3'){
    value ='3';
  } else if (choice == 'val4'){
    value ='4';
  } else if (choice == 'val5'){
    value ='5';
  } else if (choice == 'val6'){
    value ='6';
  } else if (choice == 'val7'){
    value ='7';
  }
}

function rand(){
  return Math.floor((Math.random() * 1000) + 1);
}

$('input[name="queue"]').click(function(){
  var a=rand();
  $("<div class='panel-group'>" +
    "<div class='panel panel-default'>" +
      "<div class='panel-heading'>" +
        "<h4 class='panel-title'>" +
          "<a data-toggle='collapse' href='#collapse"+a+"'>New
Exercise</a></h4>" +
        "</div>" +
      "<div id='collapse"+a+"' class='panel-collapse collapse'>" +
        "<div class='panel-body' style='padding:10px;'>" +
```

```

        "<div class='form-group'>" +
            "<label for='description'>Exercise
Description</label><br/>" +
            "<select
name='exerciseList["+value+"][]'>" +
            "<?php foreach ($exerciseList as $key =>
$item){ ?>
            "<option value='<?php echo $key;
?>'><?php echo $item; ?></option>" +
            "<?php } ?>
            "</select>" +
        "</div>" +
        "<div class='form-group'>" +
            "<label for='description'>Exercise
Description</label><br/>" +
            " <a class='card-link'
href='./exercise/'>Exercise Details</a><br/>" +
            "</div>" +
        "<div class='form-group'>" +
            "<label for='reps[]'>Number of
Reps</label><br/>" +
            "<input type='text'
name='reps["+value+"][]' size='15'> <br/>" +
            "</div>" +
        "<div class='form-group'>" +
            "<label for='sets[]'>Number of
Sets</label><br/>" +
            "<input type='text'
name='sets["+value+"][]' size='15'> <br/>" +
            "</div>" +
        "<div class='form-group'>" +
            "<label for='weight[]'>Weight
(kg)</label><br/>" +
            "<input type='text'
name='weight["+value+"][]' size='15'> <br/>" +
        "</div></div></div></div>").appendTo(".cardDay"+value);
    })
</script>

```

2.2.18 Trainer Finder (Logged in as Client/User)

CLIENT ID	NAME	USERNAME	EMAIL	REQUEST
1	Daniel Hayden	Siedro	dan@siedro.com	Request Trainer

Figure.18 Trainer Finder page

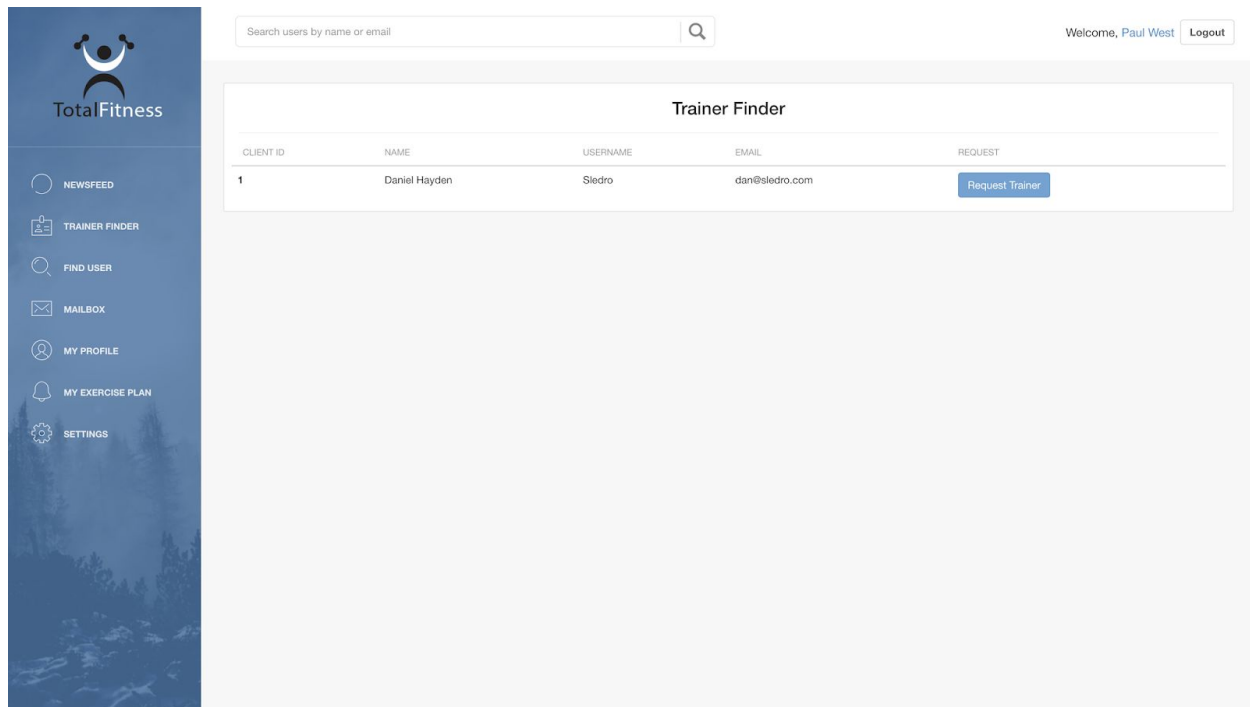
Purpose

The purpose of this page is to allow a user to find a new trainer and send them a request to become their own trainer.

Functionality and summary of work carried out

The main files used to create and manage this page are the TrainerFinder model, TrainerFinderController and the TrainerFinder>view.blade.php view.

2.2.19 My Exercise Plan (Logged in as Client/User)



The screenshot shows the TotalFitness web application interface. At the top left is the TotalFitness logo. Below it is a navigation menu with icons and labels for: NEWSFEED, TRAINER FINDER, FIND USER, MAILBOX, MY PROFILE, MY EXERCISE PLAN, and SETTINGS. At the top right, there is a search bar with the placeholder text "Search users by name or email" and a magnifying glass icon. To the right of the search bar, it says "Welcome, Paul West" and a "Logout" button. The main content area is titled "Trainer Finder" and contains a table with the following data:

CLIENT ID	NAME	USERNAME	EMAIL	REQUEST
1	Daniel Hayden	Siedro	dan@siedro.com	Request Trainer

Figure.19 My exercise plan page

Purpose

The purpose of this page is to allow a user to find a new trainer and send them a request to become their own trainer.

Functionality and summary of work carried out

The main files used to create and manage this page are the ExercisePlans model, ExercisePlansController and the ExercisePlans>viewPlanClient.blade.php view.

2.2.20 View Exercise

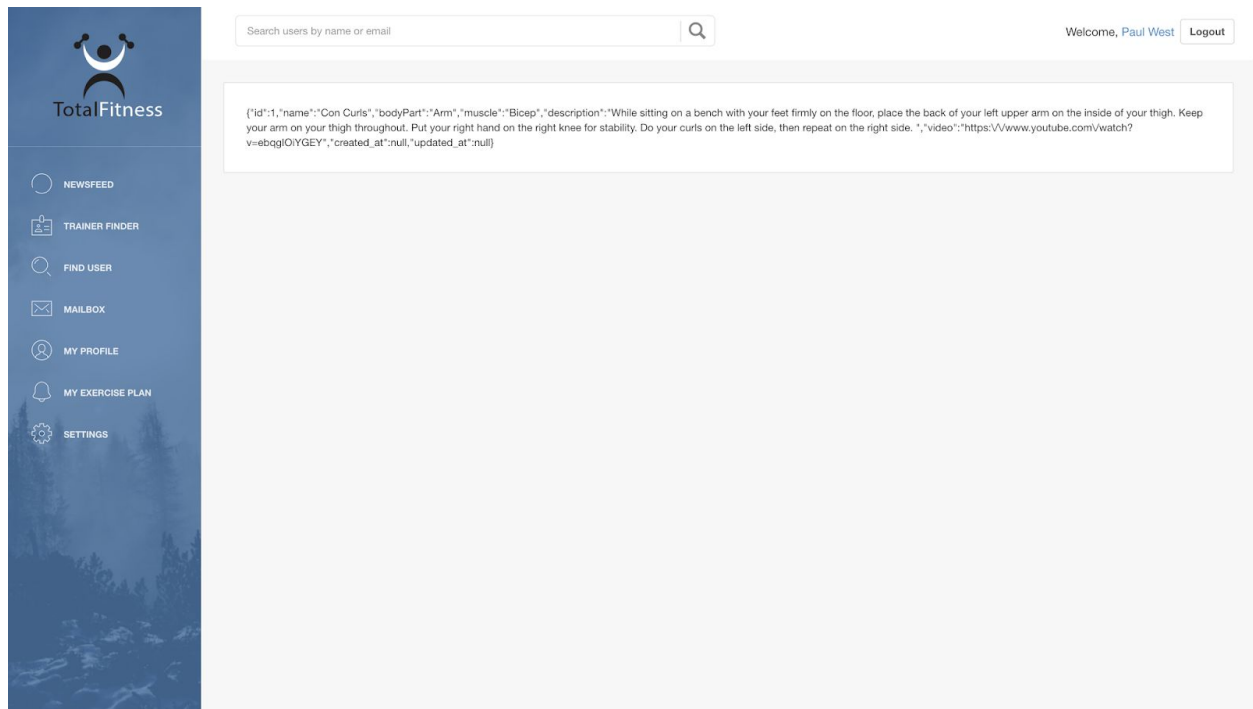


Figure.20 View exercise page

Purpose

This page is displayed when a user clicks on any of the numerous ‘exercise details’ links located in their exercise plans. In every plan where an exercise is mentioned, the details of that exercise can be viewed.

Functionality and summary of work carried out

The main files used to create and manage this page are the Exercise model, ExerciseController and the Exercise>viewExercise.blade.php view.

Note that we have the correct exercise details being successfully passed from the controller to the view. We have not yet had the time to design this page but that will be done before the program demonstration date. The youtube video links will be embedded and this page will give the user a better description of the exercise and show them how to perform the exercise correctly.

2.3 Mobile Application Framework

For the mobile application we used Ionic[6] as planned. Ionic lets you code in Angular and Typescript and then package that code to be use natively on Android, Ios and Windows devices. However instead of creating an entire new application we use Ionic to display a website in a native application using Ionics web viewer. This gives the impression that the user is using a fully native application when really they are viewing a website though the applications web viewer. This downside to this is that the user will not receive a full native experience as some buttons and menus will be displayed in html rather than the mobile operating systems native user interface components. The big advantage to using this type of application is we need to write significantly less code in a mobile applications code base. With the limited time that we have to complete this project time is golden so the pros definitely outweigh the cons in this case.

2.4 Mobile Application Framework Overview

All of the code that was developed for the mobile application can be viewed in the Technical Manual however here is the main file that routes the web viewer to display a web application that is located on a web server.

```
import { Component } from '@angular/core';
import { NavController } from 'ionic-angular';
import { InAppBrowser } from '@ionic-native/in-app-browser';

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})

export class HomePage{
  constructor(public navCtrl: NavController,private iab: InAppBrowser /* 2 */) {
    this.iab.create('http://totalfitness.host', '_self', {location:'no'}); /*3*/
  }
}
```

2.5 Mobile Application Screenshots

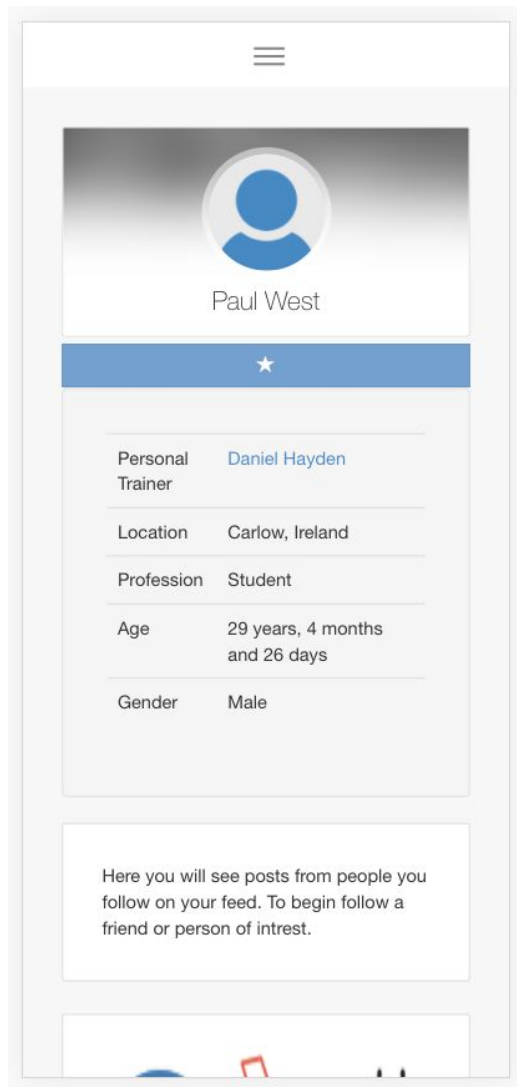


Figure 21 User Profile page mobile

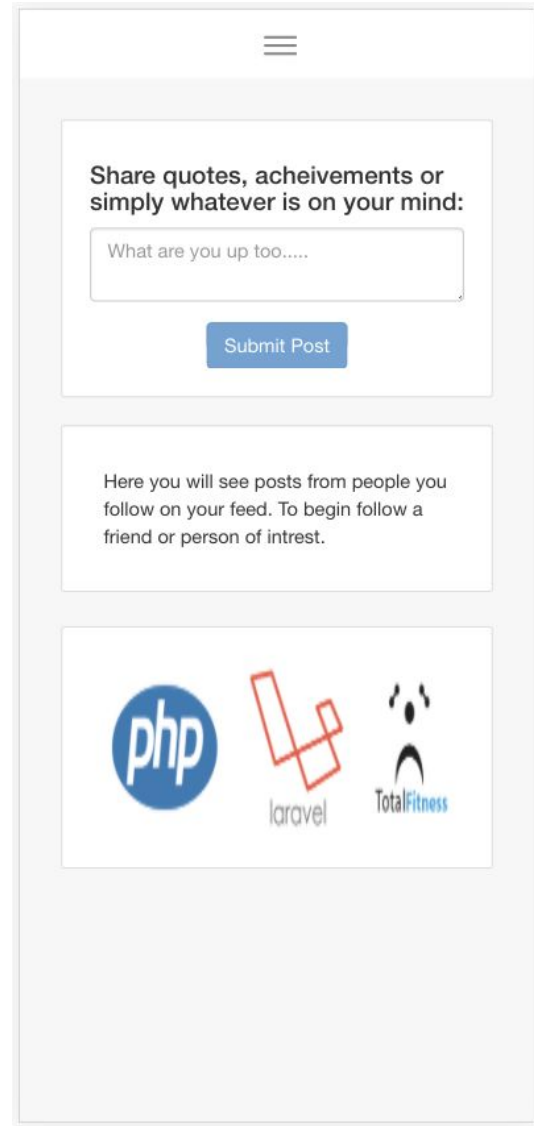


Figure 22 Newsfeed Page mobile

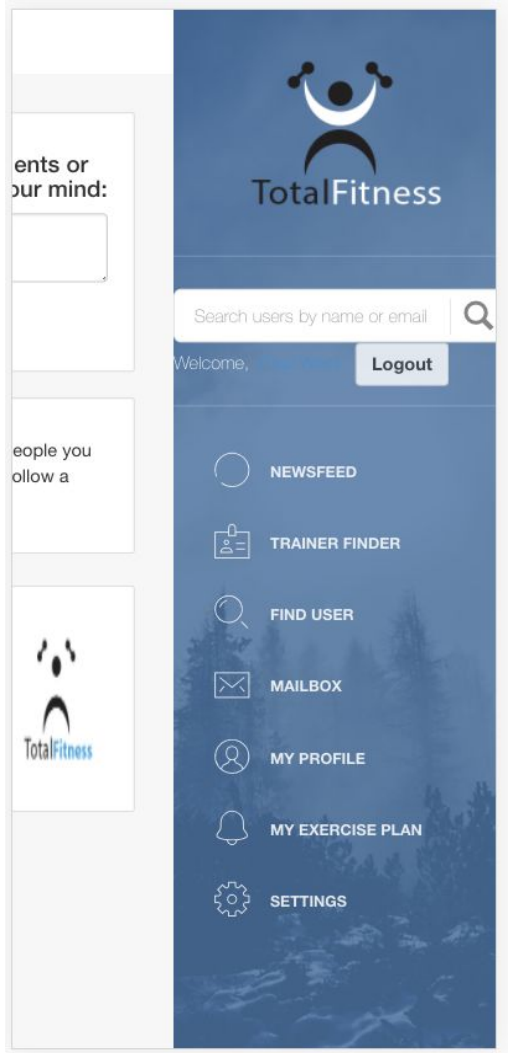


Figure 23 Menu mobile

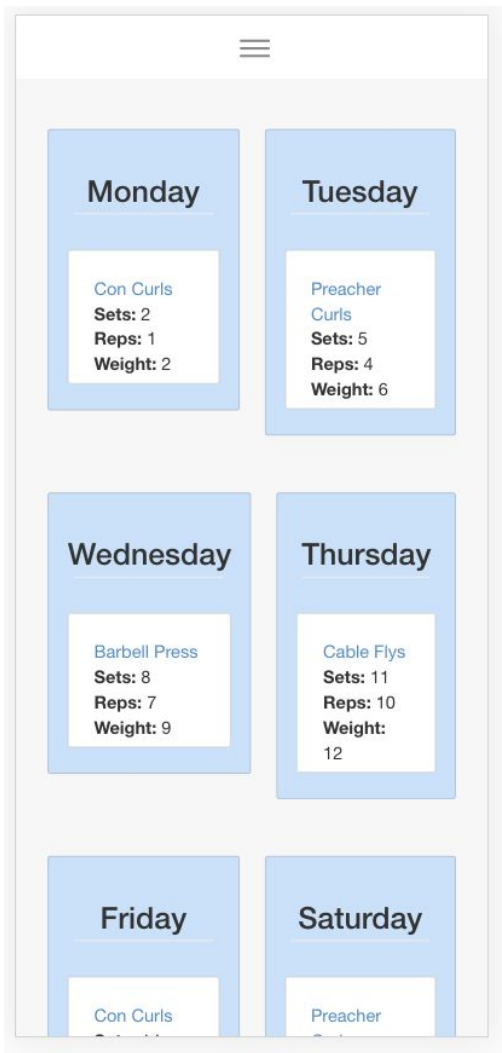


Figure 24 My Training Plan page mobile

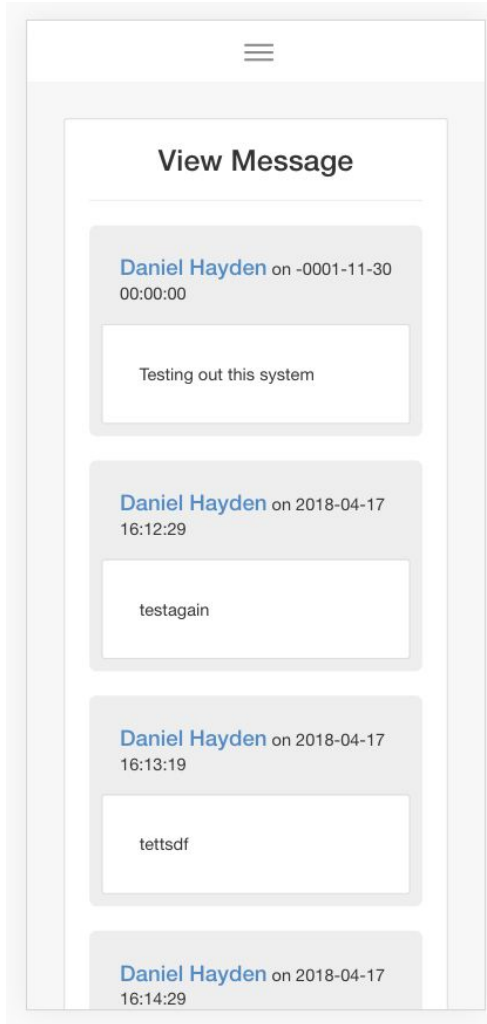


Figure 25 Private messages mobile

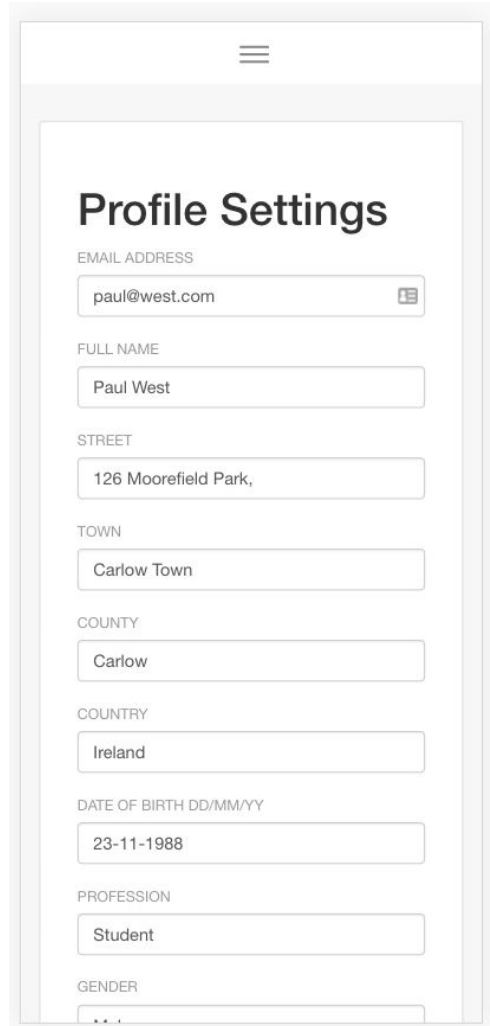


Figure 26 Settings page mobile

2.6 Database



3.0 Description of Conformance to Specification and Design

There has been a number of changes to the project since writing a functional specification and design document. The main change is the move from Google's Firebase to a MySQL database. While undergoing development to the project, we discovered that there is no official support, documentation or API to allow us to use PHP with Google's Firebase Database System. However we did find an unofficial API located on github. We began developing a php application using this API only to discover a few weeks later that the API is still under development and is very buggy. It was a tough choice but after about a month of development we decided to start over using a mySQL database.

The fact that we were now using a mySQL database and starting over allowed us to look at various PHP Frameworks and this is when we chose Laravel [1].

As we were now learning Laravel , development time was much slower then what we had anticipated when creating out documents. It was a couple of months later when we realised that we could not get this web application finished and and then develop a totally separate Ionic mobile application. We thought back to a third year Software Engineering module when we were learning about different types of mobile applications such as Native, Hybrid and Web based. We then decided that using a hybrid approach would greatly speed up a development time and allow us to have a semi-native mobile application

In summary we decided to swap out a Firebase database with a mySQL database and implement laravel and we decided to swap a fully coded ionic application for a hybrid application.

Although these are quite drastic changes from a original documents, we feel that the end product would benefit greatly from these changes in terms or progress and also quality.

4.0 Description of Learning

4.1 Technical

Over the past eight months working on this project, We have acquired a vast amount of new technical skills. We have gained a much better understanding of the PHP programming language and what can be achieved with it. As said above, using Laravel was a great experience and we look forward to creating my next project with it. We learned how to properly use the Model View Controller design pattern and which code belong in which section of the MVC pattern. We have learned a huge amount about the Ionic Framework and what it can be used for. While learning Ionic we learned how to code in Typescript and Angular. Using Bootstrap for my interface and also having a mobile interface thought me a lot about how to create responsive bootstrap user experience. Some other technical skill that we improved on while developing this project was the use of git, mysql, phpmyadmin,

4.2 Personal

On a personal level we have also learned a lot from this development project. We learned that when using the agile methodology you project can end up quite different from what you had originally planned in your documentation. Our presentations gave us much more confidence when speaking formally in public. Another thing that this project helped us with is time management. This project was carried out over a period of eight months and during those eight months we were met with numerous deadlines. Having to meet these deadlines has definitely taught me how to manage my time better. Finally I have learned how to write documents more professionally and formally.

5.0 Review of Project

Overall we would like to consider the project a success. We successfully developed a multiple platform, secure system that allows personal trainers to create plans and manage their clients. We also decided to add in some extra features such as private messaging, trainer finder, user finder and a newsfeed .

We feel that changing the projected to Laravel was a good decision as once Laravel was learned development speed was increased. If we were to start this project again we would use the same technologies and what we ended up with here except we would have more features as we would have more time to code. We did have to remove some features from the original documents such as the exercise manager due to this lost time trying to get Firebase to work correctly with PHP. If we were to give advice to another student starting a similar project in the future we would tell them to spend time researching Laravel and php instead of Firebase and PHP.

6.0 Acknowledgements

6.1 Dr. Greg Doyle

We would like to thank Dr. Greg Doyle for providing his advice whenever it was required. As my project supervisor, Greg played a key role in helping me get my documents in order and advised me on the direction of my project.

6.2 Dr. Joseph Kehoe

We would also like to thank Joseph for providing us with useful information and details about the project.

7.0 References

- [1] Laravel. 2018. The PHP Framework For Web Artisans . [ONLINE] Available at: <https://laravel.com>. [Accessed 17 April 2018].

- [2] Composer. 2018. Dependency Manager for PHP. [ONLINE] Available at: <https://getcomposer.org/>. [Accessed 17 April 2018].

- [3] Laravel. 2018. Docs - Artisan Console . [ONLINE] Available at: <https://laravel.com/docs/5.6/artisan>. [Accessed 17 April 2018].

- [4] Laravel. 2018. Docs - Eloquent: Getting Started. [ONLINE] Available at: <https://laravel.com/docs/5.6/eloquent>. [Accessed 17 April 2018].

- [5] Laravel. 2018. Docs - Authentication. [ONLINE] Available at: <https://laravel.com/docs/5.6/authentication>. [Accessed 17 April 2018].

- [6] Ionic. 2018. Docs - Build Amazing Native Applications. [ONLINE] Available at: <https://ionicframework.com/>. [Accessed 17 April 2018].