



UNIVERSITY OF CYPRUS
DEPARTMENT OF COMPUTER SCIENCE



EPL449 - Software Engineering Professional Practice

Testing and Implementation Documentation

Ironsky-Fitness App

Michalis Kaili

Andreas Savva

Sotiris Loizidis

Gregory Gregoriades

Avgerinos Sokratous

Contents

EPL449 - Software Engineering.....	1
Login	3
Trainer	5
Home	5
Subscriptions-Due:	6
Announcements	7
Statistics.....	8
Create Class:.....	10
Delete class:.....	12
Client Data:	13
Client	14
Programs:.....	14
Announcements	15
Nutritional Calculator:.....	16
1RM Calculator:.....	17
Enrollment:	19
MyAccount:.....	21
Contact Us.....	23
Memberships:.....	24
Automated Testing:.....	27
Security Techniques/Measures:.....	27
Additional Notable Work.....	28
Navbar:	28
Database Connection:	28
Cronjobs:.....	29
Physical Sign-In:.....	29
Client in the center of development	29
Web hosting.....	29

Login

To start we check the credentials the user has provided. These credentials have to be the users email (username) and his/her password. If the credentials are incorrect and do not match any entry in the database then the user is presented with an error message and is allowed to try again.

Welcome to Ironsky

Be strong, brave and humble

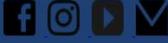


Email*

Password*

Sign-in

[Forgot password](#)



By proceeding you also agree to the [Terms of Service](#) and [Privacy Policy](#)

Welcome to Ironsky

Be strong, brave and humble

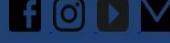


wrong username & password

.....

Sign-in

[Forgot password](#)



By proceeding you also agree to the [Terms of Service](#) and [Privacy Policy](#)



Email or Password is Invalid

OK

When the user provides correct credentials then it is established if those credentials belong to a trainer or client. This is necessary because trainer and client are shown different versions of the site with differences in functionalities and privileges.

```
$email = $_POST["Email"];
$password = $_POST["Password"];

$auth = FALSE;
$authTrainer = FALSE;
$query = "SELECT * FROM Customer WHERE Email='$email'";
$result = mysqli_query($conn, $query) or die("Could not connect database " . mysqli_error($conn));

if (!$result) {
    echo "wrong input";
    printf("Error: %s\n", mysqli_error($conn));
    exit();
}
$row = mysqli_fetch_assoc($result);

if (password_verify($password, $row['Password'])) {
    session_start();
    $_SESSION["email"] = $email;
    $_SESSION['auth'] = TRUE;
    session_write_close();
}
```

IRON SKY
FITNESS

Register New Customer

Name

First Name

Surname

Last name

Email

Email

Address

Address

Phone

Phone Number

Membership --Please Select--[Register](#)

(Trainers interface)

IRON SKY
FITNESS

Friday 01/05/2020

Thursday 30/04/2020

Wednesday 29/04/2020

Tuesday 28/04/2020

Monday 27/04/2020

Friday 24/04/2020

Thursday 23/04/2020

(Clients interface)

Trainer

Home

In the trainer home page the user is provided with two options, to register a new user or a new trainer. In each case all fields must be filled in or an error message is presented prompting the user to fill in the empty fields. Also it is not allowed to enter a new client or trainer that already exists in the database, in this case an error message informs the user that the registration failed because the credentials provided are already in the database. When a new registration is completed successfully the user is shown a success message.

The figure consists of three parts. The top part shows a registration form for a 'Customer'. It has fields for 'Name' (with 'First Name' populated) and 'Surname' (empty). A validation message 'Please fill out this field.' is shown above the empty 'Surname' field. The middle part shows a similar form for a 'Trainer', with 'Name' (containing 'Michalis Kaili') and 'Email' (empty) fields. A validation message 'Please fill out this field.' is shown above the empty 'Email' field. The bottom part shows two modal dialogs. The left dialog, titled 'This email already exists', contains the message 'Try a different one.' and an 'OK' button. The right dialog, titled 'Success!', contains the message 'User registered successfully' and an 'OK' button.

Register New Customer

Name

First Name

Surname

Please fill out this field.

Register New Trainer

Name

Michalis Kaili

Email

Email

Address

Please fill out this field.

This email already exists

Try a different one.

OK

Success!

User registered successfully

OK

```

$query = "SELECT * FROM Customer";
$result = mysqli_query($conn, $query) or die("Could not connect database ".mysqli_error($conn));

while($row = mysqli_fetch_assoc($result)) {
    $emaildb = $row['Email'];

    if($email === $emaildb) {

        echo "<script>
        swal({
        title: 'This email already exists',
        text: 'Try a different one.',
        type: 'error',
        showConfirmButton: true
        }, function(){
            window.location.href = 'http://cproject.in.cs.ucy.ac.cy/ironsky/winter19.team15/mainTrainer.php';
        });
        $('.sweet-overlay').css('background-color','#1E4072');

        </script>";
        exit();
    }
}

```

Subscriptions-Due:

In subscriptions-due page the user is shown all the memberships that have expired, if no memberships have expired the page is empty. For the expired memberships the user is able to renew them with one of the options provided. All tabs are generated dynamically from JavaScript.

IRONSKY FITNESS

Expired Memberships

Michalis Kaili Kaili

EXPIRED IN:
2020-05-01

[Renew](#)

- Unlimited 3 Months
- 3 Sessions / Week 3 Months
- Open Gym 3 Months
- Grow Strong 3 Months
- We Move 3 Months
- Unlimited**
- 3 Sessions / Week
- Open Gym
- Grow Strong
- We Move

IRONSKY FITNESS

Expired Memberships

Announcements

The trainer can create, edit and delete announcements concerning the Gym's working hours, general information and tips, or important information about important situations much like the recent epidemic. It has a specific tab on both user sides of the application. The trainer can: create, in turn update an announcement, delete it, and view all announcements.

The figure consists of three screenshots of a mobile application interface for managing announcements. The top two screenshots show the 'Announcements' screen, while the bottom one shows a detailed view of a specific post.

Screenshot 1: Create Announcement Screen

This screen shows a form for creating a new announcement. It includes fields for 'SUBJECT*' (labeled 'Concerning Wednesday program!'), 'MESSAGE*' (containing the text 'Do not let your ego dictate your form! This is a difficult workout regardless of weight'), and a green 'POST ANNOUNCEMENT' button.

Screenshot 2: All Posts Screen

This screen displays a list of announcements under the heading 'All Posts'. Each post includes the date ('May 05, 2020'), subject ('Concerning Wednesday program!', 'Hello', 'Personal Training Sessions Available!'), message preview, and 'DELETE' and 'EDIT' buttons.

Screenshot 3: Update Announcement Screen

This screen shows the details of a selected announcement for modification. It includes fields for 'SUBJECT *' (labeled 'Concerning Wednesday program!') and 'MESSAGE *' (containing the text 'Please be considerate on your weight selection. This is a difficult workout regardless of weight'), along with a green 'UPDATE ANNOUNCEMENT' button.

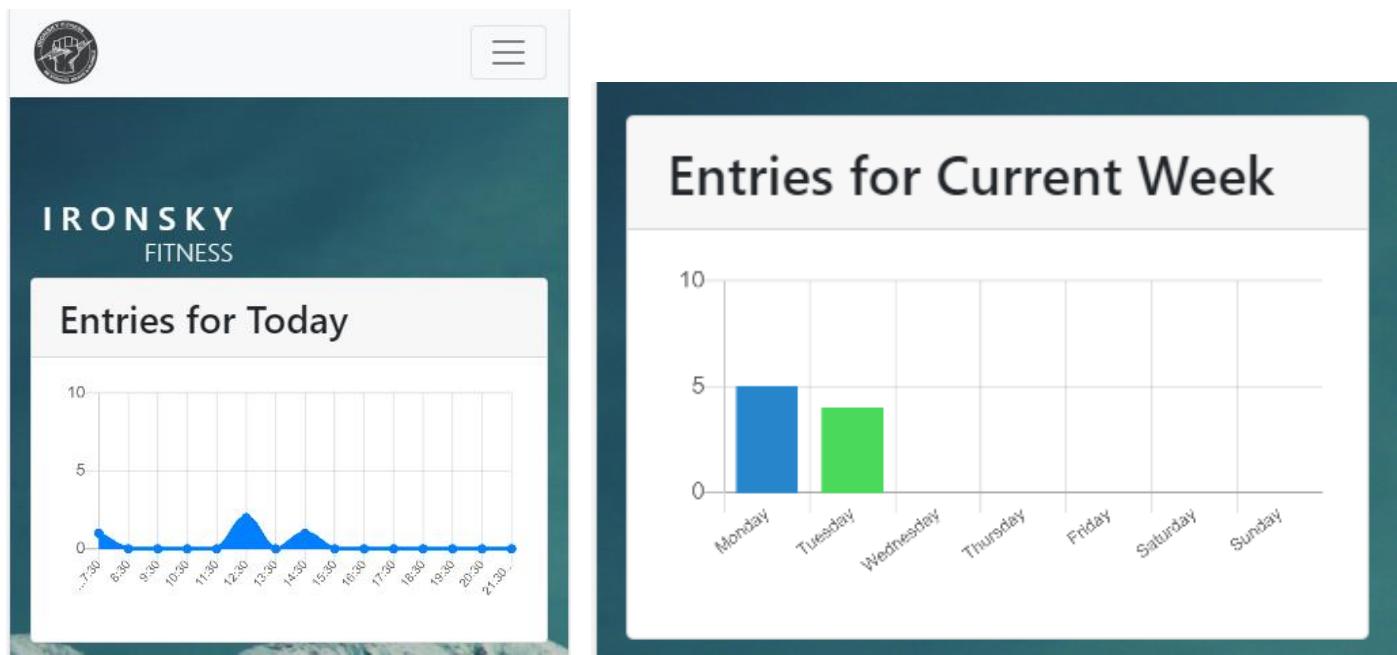
This screenshot provides a detailed view of an announcement post from May 05, 2020. The post subject is 'Concerning Wednesday program!' and the message text is 'Please be considerate on your weight selection. This is a difficult workout regardless of weight'. The post has a timestamp of 'May 05, 2020' at the bottom.

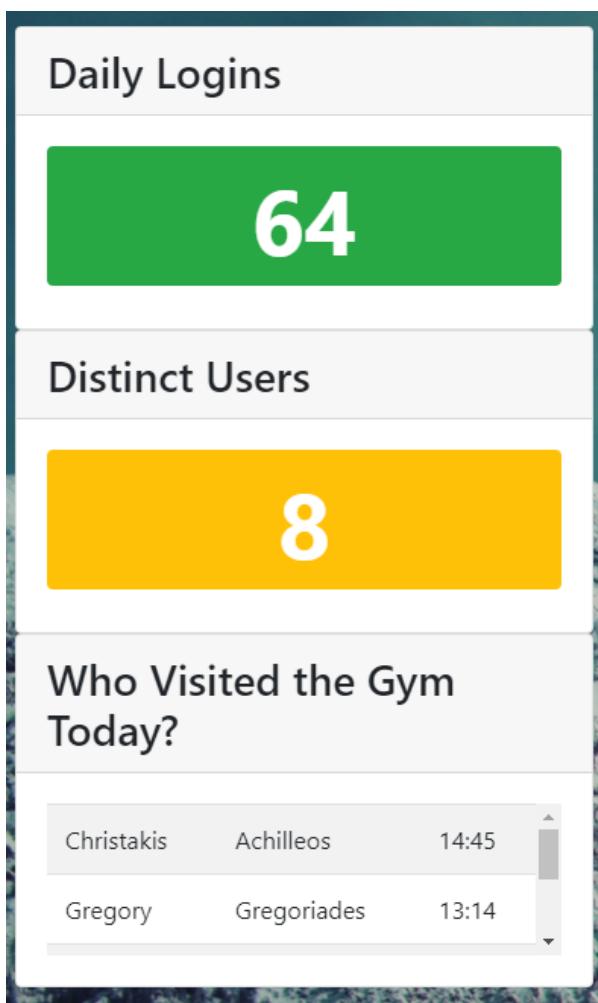
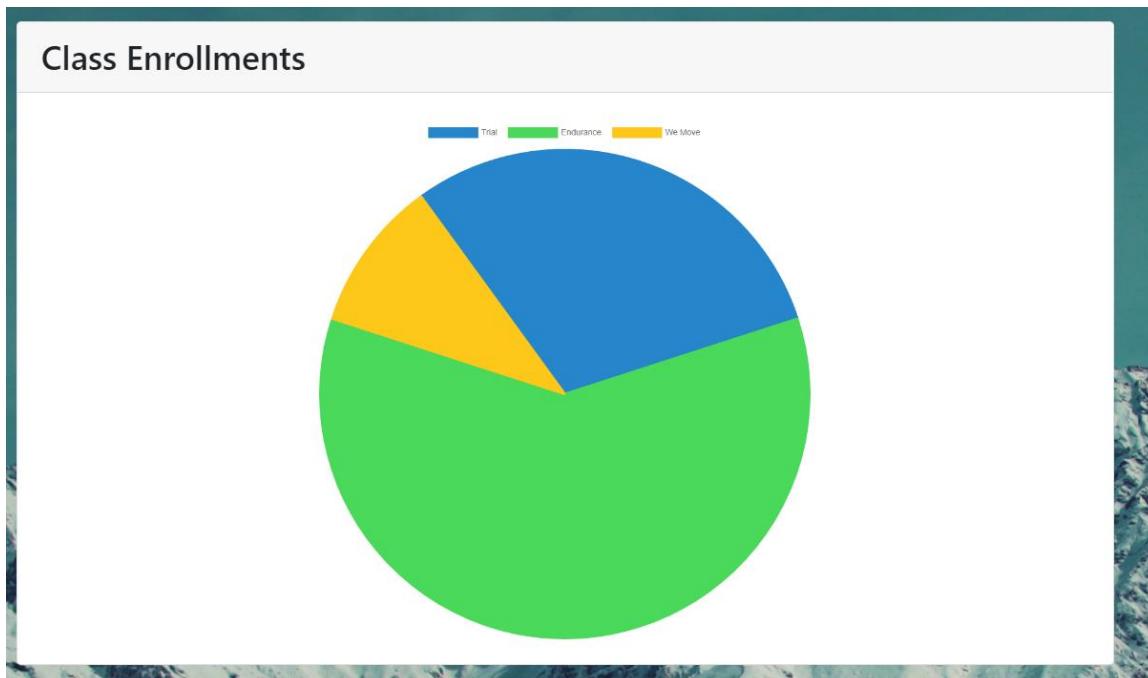
Statistics

One of our nicest and most useful functions is the Statistics tab. A plethora of statistics is visualized in a user-friendly manner to showcase different metrics concerning:

- The amount of people who have physical signed-in the gym facility for the current day. This is done via GPS location tracking with the help of the https protocol
- A bar graph showcasing the same metric but for all days of the current week. At later stages we could ass a monthly graph to mine some insightful trends.
- Class enrollments for the current week. This way the trainer can determine the most popular classes and change them accordingly.
- The number of daily logins to the client page
- The number of distinct users logging into the client page
- The names of specifically of the clients that entered the gym today.

Future work could extend this section indefinitely. You can never have enough statistics.





Create Class:

In the create class page the user can create new classes and place them in the desired days and times he/she wishes. To do so the user is required to insert a name for the class, the number of available places it has and a brief description. If any of these fields are left empty an error message will remind the user that he/she must fill them in order to create the new class. Only one class can be placed in each time slot, if a class already exists no other class can be created in the same slot and it cannot be overwritten. The old class must be deleted before a new one is placed in the same slot.



Create New Class

Class Name
 ! Please fill out this field.

Number of places:
 ! Please fill out this field.

Hours	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7:30-8:15	Strength		Strength		Strength	Trial
8:15-9:00	DIMOIRIA EPIDIKSEWN					Trial
9:00-12:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM
12:30-15:30	CLOSED	CLOSED	CLOSED	CLOSED	CLOSED	
15:30-17:15	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	
17:15-18:00	DIMOIRIA EPIDIKSEWN					
18:00-18:45	DIMOIRIA EPIDIKSEWN					
18:45-19:30	DIMOIRIA EPIDIKSEWN					
19:30-20:15	DIMOIRIA EPIDIKSEWN					
20:15-21:00						
20:15-21:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	

Brief Description

Create

```

obj.forEach(function(obj) {
    $("table td").each(function () {
        if ($(this).attr("data-id") != "1"){
            if(($(this).attr("data-position") === obj.classIndex)&&(!($(this).hasClass("alreadyChecked")))){
                // obj["classIndex"] = $(this).attr("data-position");
                // obj["className"] = obj.className;
                document.getElementById(($(this).attr("data-position"))).innerHTML=obj.className;
                $(this).toggleClass("alreadyChecked");
            }
        }
    });
});

$("table td").click(function () {
    if (($(this).attr("data-id") != "1")&&(!($(this).hasClass("alreadyChecked"))))
        $(this).toggleClass("checked");
});
var positions = [];
$("#submit").on("click", function (e) {
    var className = $("#input1").val();
    var nPlaces = $("#input2").val();
    var description = $("#input3").val();
    if (!className || !nPlaces || !description) {
        return;
    } else {

        $("table td").each(function () {
            if ($(this).hasClass("checked")){
                positions.push($(this).attr("data-position"));
            }
        });

        $.ajax({
            url: "php/postClass.php",
            type: "POST",
            data: { className:className,positions: positions,nPlaces:nPlaces,description:description },
            success: function(data){
            }
        });
    }
})
}

```



Create New Class

Class Name

Number of places:

Hours	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7:30-8:15	Strength		Strength		Strength	Trial
8:15-9:00	DIMOIRIA EPIDIKSEWN					Trial
9:00-12:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM
12:30-15:30	CLOSED	CLOSED	CLOSED	CLOSED	CLOSED	
15:30-17:15	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	
17:15-18:00	DIMOIRIA EPIDIKSEWN					
18:00-18:45	DIMOIRIA EPIDIKSEWN					
18:45-19:30	DIMOIRIA EPIDIKSEWN					
19:30-20:15	DIMOIRIA EPIDIKSEWN					
20:15-21:00						
20:15-21:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	

Brief Description

Create

Delete class:

In the delete class page the user can delete any class by simply selecting it and pressing the delete button. The user can delete multiple classes at once. It works the same way as the create class but in the opposite way which means you can only select slots that are already filled.

Delete Classes

Hours	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7:30-8:15	Strength		Strength		Strength	Trial
8:15-9:00	DIMOIRIA EPIDIKSEWN					Trial
9:00-12:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM
12:30-15:30	CLOSED	CLOSED	CLOSED	CLOSED	CLOSED	
15:30-17:15	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	
17:15-18:00	DIMOIRIA EPIDIKSEWN		new class name			
18:00-18:45	DIMOIRIA EPIDIKSEWN		new class name			
18:45-19:30	DIMOIRIA EPIDIKSEWN		new class name			
19:30-20:15	DIMOIRIA EPIDIKSEWN		new class name			
20:15-21:00						
20:15-21:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	

Delete**Delete Classes**

Hours	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7:30-8:15	Strength		Strength		Strength	Trial
8:15-9:00	DIMOIRIA EPIDIKSEWN					Trial
9:00-12:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM
12:30-15:30	CLOSED	CLOSED	CLOSED	CLOSED	CLOSED	
15:30-17:15	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	
17:15-18:00	DIMOIRIA EPIDIKSEWN					
18:00-18:45	DIMOIRIA EPIDIKSEWN					
18:45-19:30	DIMOIRIA EPIDIKSEWN					
19:30-20:15	DIMOIRIA EPIDIKSEWN					
20:15-21:00						
20:15-21:30	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	OPEN GYM	

Delete 2

Client Data:

In the client data page, the user can use two functionalities. The first functionality is the retrieve client data. Here the trainer can retrieve all of the data stored for a client upon the clients' request. The user must fill in the two necessary fields with the clients' credentials, the email and password is required in order to make sure that only the client can receive his data, it is necessary safety measure. After the credentials are placed they will be checked to see if they are valid. If the credentials provided are not valid an alert message will be show informing the user. This can mean that the credentials were wrong or that this user does not exist in the database. If the credentials are valid then an email will be sent to the user containing all the information that concerns them that were found in the database.

The screenshot shows a web application interface. On the left, there is a form titled "Retrieve Client Data". It has two input fields: "Email" containing "wrongEmail@gmail.com" and "Password" containing "*****". Below the fields is a green "Retrieve" button. On the right, a modal dialog box is displayed with the URL "http://cproject.in.cs.uct.ac.cy/ironsky/winter19.team15/php/retrieveClientData.php". The dialog contains the text "FAILED TO VERIFY USER" and an "OK" button. The background of the application shows a dark blue header with the text "keras_meetup" and "cproject.in.cs.uct.ac.cy says".

If the client wants all of their data delete permanently from the database then the user follows the same procedure but this time in the delete client data portion of the page. After pressing the delete button the same verification process takes place and if the user is verified then all of their data is deleted an email is sent to them informing them of the success of their request.

```
if (password_verify($password,$row['Password']))
{
    $myfile = fopen("testData.txt", "w") or die("Unable to open file!");

    $cuid="SELECT Customer_ID FROM Customer WHERE Email = '$email'";
    $res=mysqli_query($conn,$cuid);
    $row = mysqli_fetch_assoc($res);
    $id=$row['Customer_ID'];

    $query = "SELECT * FROM Customer WHERE Customer_ID = '$id' ";
    $res=mysqli_query($conn,$query);
    $data = mysqli_fetch_assoc($res);
    $fieldInfo = mysqli_fetch_fields($res);
    fwrite($myfile, "Client Details: \n");
    foreach ($fieldInfo as $field )
    {
        //echo "<script> alert($field->table); </script>";
        fwrite($myfile,$field->name . " => " . $data[$field->name] . "\n");
    }
}
```

Client

Programs:

Once client signs into their account they are greeted by the programs page. Here the user can see the gyms program of the day for the last seven days. The latest program is open by default for the users' convenience.

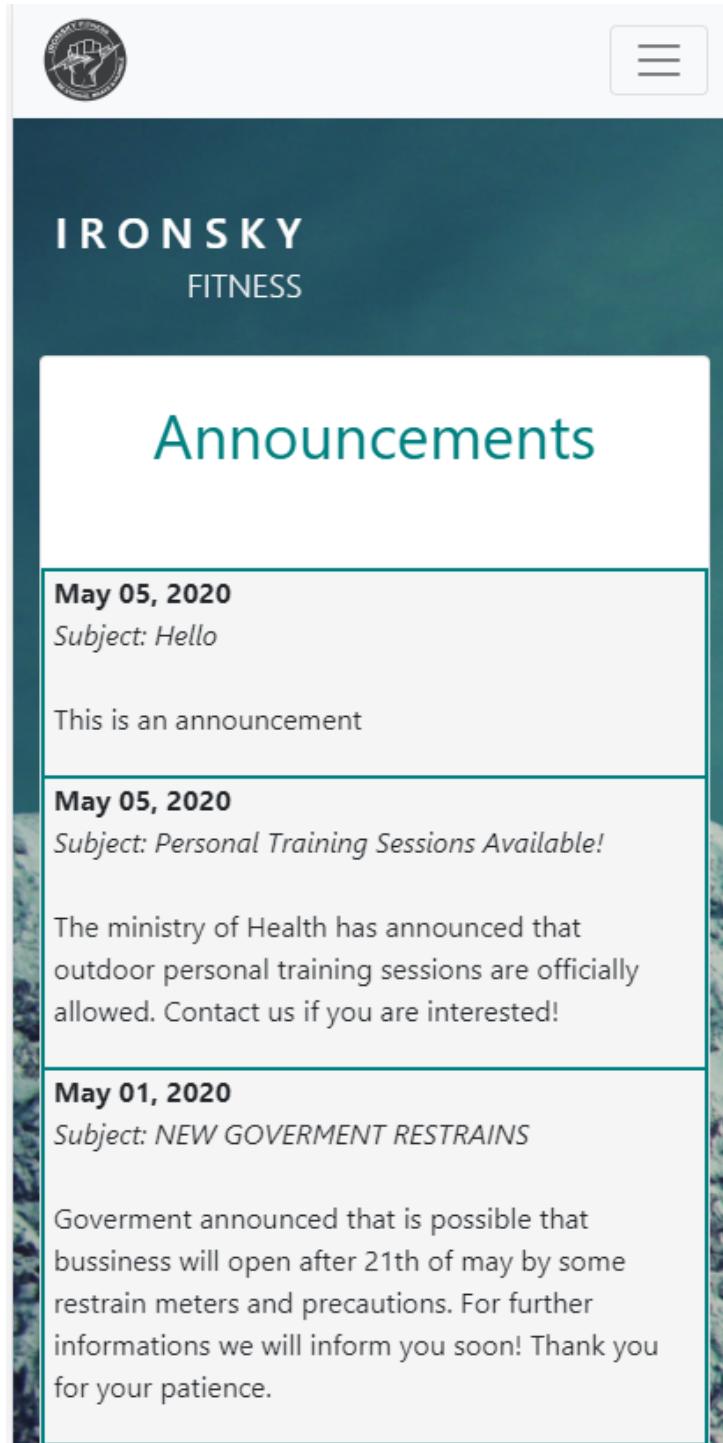
```

$( document ).ready(function()
{
  setTimeout(function()
  {
    var i = 0;
    var x = document.getElementsByClassName("BlogItem-title");
    console.log(x);
    var y = document.getElementById("list-n1-list");
    if(!dayEmpty0)
    {
      y.innerText = x[i].innerText;
      i = i + 1;
    }
    else
    {
      y.innerText = " ";
    }
  })
})

```

Announcements

The client side of the announcement's functionality, the user can simply view the announcement that the trainers put up.



The image shows a mobile application interface for "IRONSKY FITNESS". At the top, there is a logo on the left and a menu icon (three horizontal lines) on the right. The main header reads "IRONSKY FITNESS". Below this, a large blue box contains the word "Announcements" in white. Three announcement cards are listed vertically:

- May 05, 2020**
Subject: Hello
This is an announcement
- May 05, 2020**
Subject: Personal Training Sessions Available!
The ministry of Health has announced that outdoor personal training sessions are officially allowed. Contact us if you are interested!
- May 01, 2020**
Subject: NEW GOVERNMENT RESTRAINS
Government announced that it is possible that business will open after 21st of May by some restraint meters and precautions. For further informations we will inform you soon! Thank you for your patience.

Nutritional Calculator:

In the nutritional calculator the user can insert information concerning their physical attributes as well as their future goals they wish to achieve. By combining this information and executing some calculations the page presents the user with information and guidelines to help the user achieve their goal. This calculator will display an error message if the information inserted is illogical or unrealistic. This is done to protect the user from possible mistakes they may have made or to protect from malicious behavior.

<p>Calculation of BMR and Calories</p> <p>Gender: <input checked="" type="radio"/> Male <input type="radio"/> Female</p> <p>Weight <input type="text" value="4521"/></p> <p>Height <input type="text" value="321"/></p> <p>Age <input type="text" value="22"/></p> <p>Lifestyle <input style="width: 200px; height: 20px; border: none; border-bottom: 1px solid #ccc; padding: 2px; margin-bottom: 5px;" type="text" value="I am moderately active (moderate exercise or sp"/></p>	<p>Calculation of BMR and Calories</p> <p>Gender: <input checked="" type="radio"/> Male <input type="radio"/> Female</p> <p>Weight <input type="text" value="100"/></p> <p>Height <input type="text" value="185"/></p> <p>Age <input type="text" value="25"/></p> <p>Lifestyle <input style="width: 200px; height: 20px; border: none; border-bottom: 1px solid #ccc; padding: 2px; margin-bottom: 5px;" type="text" value="I am lightly active (light exercise or sports 1-3 d"/></p>	<p>Your BMR is: 2036.25</p> <p>Body Fat %: 13.11%</p> <p>Lean Mass: 86.9 kg</p> <p>Fat Mass: 13.1kg</p> <p>Maintenance calories per day: 2800</p> <p>Calories per day to achieve goal: 1698</p>
<p>Weight Gain/Loss Goal (Optional)</p> <p>Weight Goal <input type="text" value="90"/></p> <p>Time Span <input type="text" value="70"/></p> <p><small>*Please enter realistic information</small></p>		<p>Calculate</p>
<p>Weight Gain/Loss Goal (Optional)</p> <p>Weight Goal <input type="text" value="90"/></p> <p>Time Span <input type="text" value="70"/></p>		<p>Calculate</p>

(Unrealistic information)

(Realistic information)

```

var bmr = (10*weight) + (6.25*height) - (5*age) + 5;
var lifestyle = document.getElementById('inputState').value;
var calories = Math.round(bmr * lifestyle);
var bmi = weight/((height/100)*(height/100));

if(age<18){
    var fat= Math.round((1.51 * bmi - 0.70 * age - 2.2)*100)/100;
}else{
    var fat= Math.round((1.20 * bmi - 0.23 * age - 16.2)*100)/100;
}

var fatmass= Math.round((weight * fat/100)*10)/10;
var lean= Math.round((weight * (100 - fat) / 100)*10)/10;

if(((document.getElementById('weight-goal').value)!=="")&&((document.getElementById('time-span').value)!=="")) {
    var goal= document.getElementById('weight-goal').value;
    var days= document.getElementById('time-span').value;

    if(goal>=weight){
        var dif= goal-weight;
    }else{
        var dif= -(goal-weight);
    }
    var temp = dif/0.0001296;
    var calperday = Math.round(temp/days);
    if(goal>=weight){
        var goalcal= calories + calperday;
        document.getElementById('goalcard').hidden= false;
        document.getElementById('cal-goal').innerHTML= goalcal;
    }else{
        var goalcal= calories - calperday;
        document.getElementById('goalcard').hidden= false;
        document.getElementById('cal-goal').innerHTML= goalcal;
    }
}

```

1RM Calculator:

The 1rm calculator page requires the user to insert the weight they used and the number of repetitions they accomplished. With this information some formulas are used to present to the user the predicted maximum weight they can use in a set for a different amount of repetitions between 1-10.

```

calculateRMs: function() {

    // calculate RMs 2-10 and append it to the table
    var i = 2;
    for (i; i <= 10; i++) {

        RepMaxCalc.lomrm = Math.floor(RepMaxCalc.lomonerm / (Math.pow(i, 1 / 10)));
        RepMaxCalc.brzrm = Math.floor((RepMaxCalc.brzonerm * (37 - i)) / 36);
        RepMaxCalc.eplrm = Math.floor(RepMaxCalc.eplonerm / ((1 + (i / 30))));
        RepMaxCalc.mayrm = Math.floor((RepMaxCalc.mayonerm * (52.2 + (41.9 * Math.exp(-1 * (i * 0.055)))))) / 100;
        RepMaxCalc.ocorm = Math.floor((RepMaxCalc.ocononerm / (1 + i * 0.025)));
        RepMaxCalc.watrm = Math.floor((RepMaxCalc.watonerm * (48.8 + (53.8 * Math.exp(-1 * (i * 0.075)))))) / 100;
        RepMaxCalc.lanrm = Math.floor(((RepMaxCalc.lanonerm * (101.3 - 2.67123 * i)) / 100));

        RepMaxCalc.avgrm = Math.floor((RepMaxCalc.lomrm + RepMaxCalc.brzrm + RepMaxCalc.eplrm + RepMaxCalc.mayrm +
            RepMaxCalc.ocorm + RepMaxCalc.watrm + RepMaxCalc.lanrm) / 7);
    }
}

```

Estimate your 1RM

These values are just for reference. Be careful!

I did **80** for **5** reps

RM	Average	Lombardi	Brzycki	Epley	Mayhew	O'Conner	Wathan	Lander
1RM	92	93	90	93	95	90	93	90
2RM	86	87	87	87	85	85	88	87
3RM	84	84	85	84	83	83	85	84
4RM	81	81	82	82	81	81	82	82
5RM	80	80	80	80	80	80	80	80
6RM	77	78	77	77	78	78	77	77
7RM	75	77	75	75	76	76	75	75
8RM	73	76	72	73	75	75	73	72
9RM	72	75	70	71	74	73	71	70
10RM	70	74	67	70	72	72	69	67

Enrollment:

In the enrollment page the user can select a day to view the available classes and enroll in the ones they wish if there are available positions. If the user doesn't select an error message is presented to inform them to do so.

The screenshots illustrate the enrollment process across four different days:

- Sunday:** Shows a dropdown menu set to "Sunday" and a search button. A red error message "Please select day!" is displayed next to the search button.
- Tuesday:** Shows a dropdown menu set to "Tuesday" and a search button. Below the search button, a message says "Unfortunately no classes are available".
- Wednesday:** Shows a dropdown menu set to "Wednesday" and a search button. A single class is listed: "1106 Strength Survive. 7:30 12". An "Enroll" button is present to the right of the class details.
- Thursday:** Shows a dropdown menu set to "Thursday" and a search button. A single class is listed: "1106 Strength Survive. 7:30 12". An "Enroll" button is present to the right of the class details. To the right of the "Enroll" button is a small image of a barbell.

```

var btn = document.createElement('input');
btn.type = "button";
btn.className = "btn btn-primary";
btn.value = "Enroll";
btn.id = "enroll";
btn.onclick =(function () {
    var t = document.getElementById('Classes').rows[i + 1].cells[0].innerHTML;
    $.ajax({
        url: 'php/enrollCust.php',
        type: 'POST',
        data: { id: t , datediff: diff },
        success: function (data) {
            console.log(data);
            if (data == 1) {
                var pl = document.getElementById('Classes').rows[i + 1].cells[5];
                var num = document.getElementById('Classes').rows[i + 1].cells[5].innerHTML;
                pl.innerHTML = num - 1;
                btn.parentNode.removeChild(btn);
                swal({
                    title: 'Enrollment',
                    text: 'Successfully enrolled',
                    type: 'success',
                    showConfirmButton: true
                }, function(){
                    window.location.href = 'https://www.ironsky-app.com/enroll.php';
                });
            }else{
                var reason="";
                switch(parseInt(data)) {
                    case (-1):{
                        reason="Already Enrolled";
                    }
                }
            }
        }
    })
}

//if he is allowed only 3 times, check if he has enrolled three times for the week he wishes to enroll
if($three){
    //get the week he wishes to enroll
    $sql="SELECT WEEK(DATE_ADD(CURRENT_DATE,INTERVAL ".$diff." Day)) AS WEEK";
    mysqli_query($conn,$sql);
    $result = $conn->query($sql);
    $r = mysqli_fetch_assoc($result);
    $week=$r['WEEK'];

    //get the totals of the weeks he has enrolled
    $sql="SELECT COUNT(Id),WEEK(Date) FROM Enrolled WHERE Username='".$mail."' GROUP BY WEEK(Date)";
    mysqli_query($conn,$sql);
    $result = $conn->query($sql);
    while($r = mysqli_fetch_assoc($result)) {
        if($r['WEEK(Date)']==$week){
            if($r['COUNT(Id)']==3){
                echo(-3);
                exit();
            }
        }
    }
}
}

```

After the user enrolls into a class the number of available classes is decreased and the user is given the option to leave (unenroll) from the class.

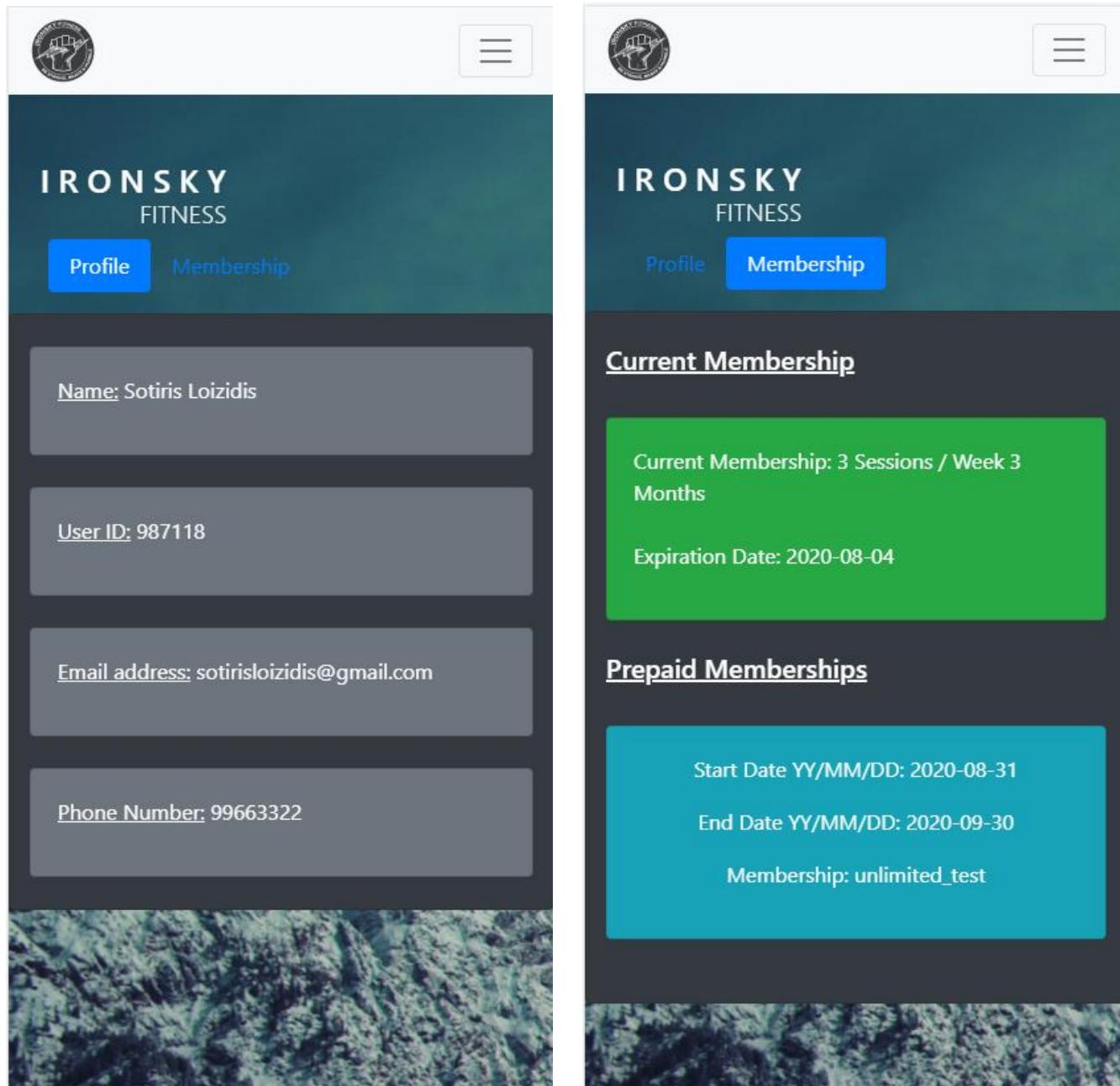
ClassID	Class	Description	Time	Number Of Places	Available Places	Enrollment
1106	Strength	Survive.	7:30	12	11	UnEnroll

Additionally the user may be rejected from enrolling in a class depending on their membership, they may have an open gym membership which does not include classes or they may have exceeded their allowed enrollments for example maximum of three in “3 sessions per week” membership”.

MyAccount:

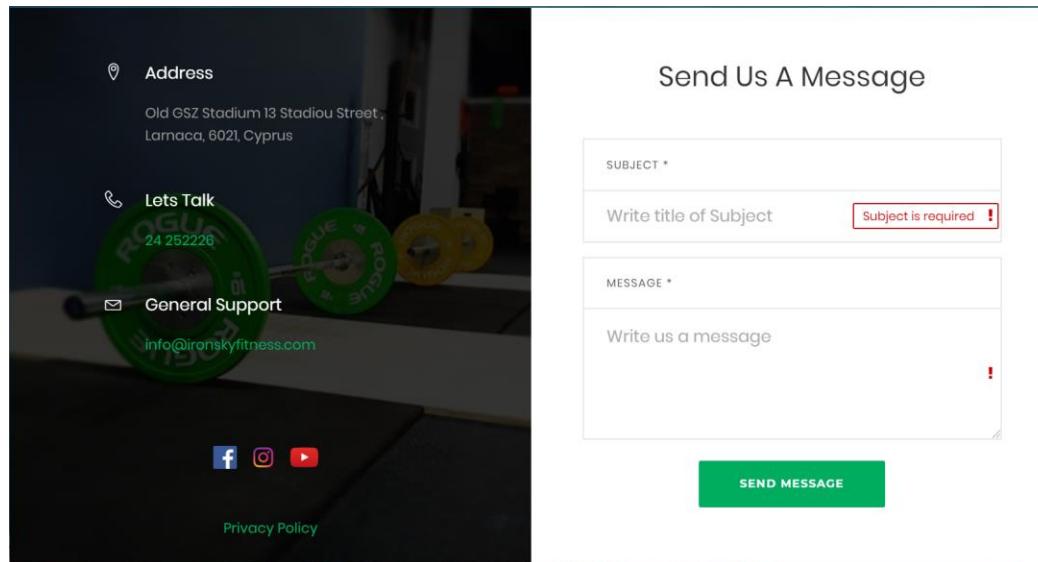
Utility function that was added upon request of beta testers which are clients of the Gym. This was not discussed as a functionality of the initial project, but it serves as a mean for the client to be able to keep track of their current Membership expiration date and type and lists future prepaid memberships that will activate once the current one expires. There is also a profile tab which simply lists the personal information of the client and essentially all the personal information that we store for them. Future work is to provide an edit

button and add tabs that let the client make a list of their personal best, track weight etc.



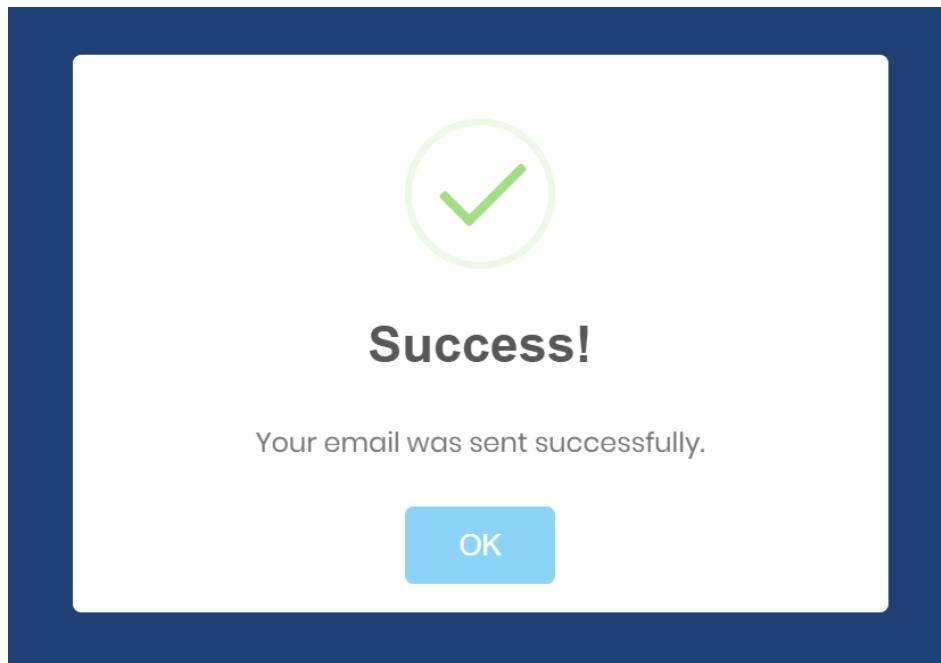
Contact Us

In the contact us page the user can send a message to the stuff of the gym by just inserting a subject and a message in the designated areas. An error message is presented to the user if one of the two fields is left empty. The message is sent using the users email so the stuff can reply to them.



The image shows a composite view. On the left, there's a dark overlay of a gym website featuring a barbell, contact details like address and phone number, and social media links. On the right, a white contact form titled "Send Us A Message" is displayed. It has fields for "SUBJECT *" and "MESSAGE *". Both fields have placeholder text: "Write title of Subject" and "Write us a message". A red error message "Subject is required!" is shown above the subject field. A green "SEND MESSAGE" button is at the bottom. The entire form is enclosed in a light gray border.

Once the user successfully sends the email a success message is shown so that the user knows that his message was successfully send.

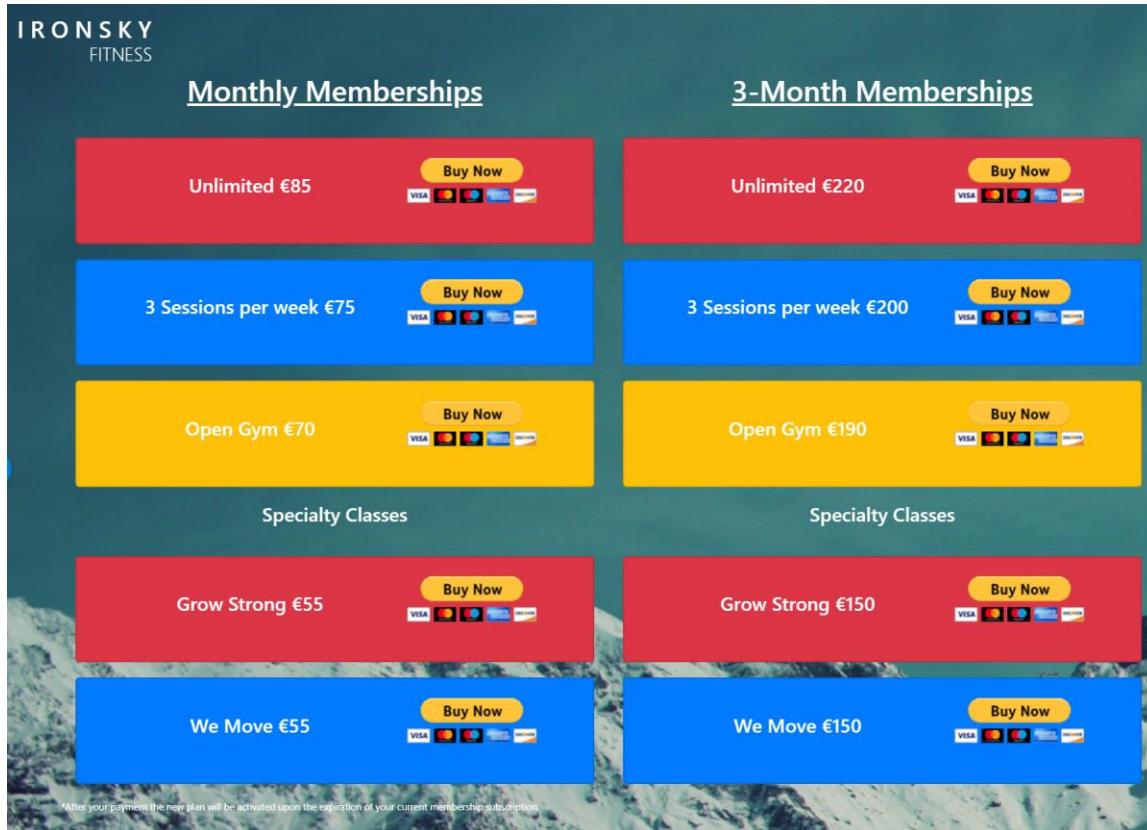


Memberships:

In the memberships page the user can see all the available memberships provided by the gym. The user can decide which membership they wish to buy depending on their needs. After deciding what they want to buy they can click on the appropriate button after which they will be taken to the PayPal page so they can proceed with the payment. After the payment is complete, they will be redirected back to the memberships page. If the user's membership is expired it will be updated to the chosen membership with a starting date equal to the previous expiration date. If the user pays for a membership before the active one expires the newly paid membership will be stored and updated accordingly once the current one expires. An extra option was added in the memberships page in order to test its functionality and security. This extra option will not be included in the final product.

The screenshot shows the 'Memberships' page for Iron Sky Fitness. The page is divided into two main sections: 'Monthly Memberships' and '3-Month Memberships'. Under 'Monthly Memberships', there are four options: 'Unlimited test €0.01', 'Unlimited €85', '3 Sessions per Week €75', and 'Open Gym €70'. Under '3-Month Memberships', there are three options: 'Unlimited €220', '3 Sessions per week €200', and 'Open Gym €190'. Each option includes a 'Buy Now' button and a list of accepted payment methods (VISA, MasterCard, American Express, etc.). The page has a background image of a snowy mountain. A small note at the bottom states: 'After your payment the new plan will be activated upon the expiration of your current membership subscription.'

(Testing memberships page)



(Final memberships page)

```

case "We Move":
{
    if ($_POST["mc_gross"] == 55 && $_POST["mc_currency"] == "EUR" && $_POST["payment_status"] == "Completed") {
        $email_to = $_POST["first_name"] . " " . $_POST["last_name"] . "<" . $_POST["payer_email"] . ">";
        $email_subject = $test_text . "Completed order for: " . $_POST["item_name"];
        $email_body = "You have renewed your membership with the " . $_POST["item_name"] . " plan." . "\r\n" . "\r\n" . "Thank you.";
        $session_email=$_POST["custom"];

        mail($email_to, $email_subject, $email_body, "From: " . $from_email_address);
    }
}
break;

```

This code represents part of the verification process to ensure that the right amount has been paid, in the correct currency for the corresponding membership and an automated email is sent to the client to inform them that the payment has been completed successfully.

```

<div class="row mx-auto card-body d-flex justify-content-center align-items-center container">
<h5 class="col-lg-7 card-title" style="font-size:20;">Unlimited test &#8364;0.01</h5>

<form action="https://www.paypal.com/cgi-bin/webscr" method="post" target="_top">
<input type="hidden" name="cmd" value="_s-xclick">
<input type="hidden" name="hosted_button_id" value="6SKFHR39AXKFS">
<input type="hidden" name="custom" value="<?=$_SESSION['email'];?>">
<input type="image" src="https://www.paypalobjects.com/en_US/i/btn/btn_buynowCC_LG.gif" border="0" name="submit" alt="PayPal - The safer way to pay online!">

</form>

```

This code segment shows the payment buttons which are generated by PayPal. We encase them in html code to format them according to our sites appearance.

```

// Reply with an empty 200 response to indicate to paypal the IPN was received correctly
header("HTTP/1.1 200 OK");

$pmpstat=$_POST["payment_status"];
$tranId=$_POST["txnid"];
require('connectDB.php');

if($paypal_ipn_status == "Completed Successfully"){
    if($pmstat=="Completed"){

        //check for duplicates
        $sql="SELECT * FROM Paypal WHERE TransactionID='$tranId'";
        $result=mysqli_query($conn,$sql);
        while($r = mysqli_fetch_assoc($result)) {
            if($r['Transaction_Status']=="Completed"){
                exit();
            }
        }

        //get customer email
        $sql="SELECT Customer_ID FROM Customer WHERE Email='".$session_email."'";
        $result=mysqli_query($conn,$sql);
        $row = mysqli_fetch_assoc($result);
        $userID=$row['Customer_ID'];

        //get duration of membership
        $sql="SELECT Duration FROM Membership_Types WHERE Type='".$membership."'";
        $result=mysqli_query($conn,$sql);
        $row = mysqli_fetch_assoc($result);
        $duration=$row['Duration'];

        //get expiry date of current membership
        $sql="SELECT ExpirationDate FROM Memberships WHERE CustomerID=".$userID;
        $result=mysqli_query($conn,$sql);
        $row = mysqli_fetch_assoc($result);
        $exp=$row['ExpirationDate'];

        //update membership
        $up="UPDATE Memberships SET Type='Unlimited',ExpirationDate=DATE_ADD('$exp',INTERVAL ".$duration." MONTH) WHERE CustomerID=".$userID;
        mysqli_query($conn,$up);

    }

    $sql="INSERT INTO Paypal(TransactionID,Transaction_Status) VALUES ('$tranId','$pmstat')";
    mysqli_query($conn,$sql);
    mysqli_close($conn);
}

```

In this code segment we perform various checks to make sure that any issues/problems that may occur are dealt with. These problems may include packet loss and duplicates. Additionally we renew the clients' membership in the database.

Automated Testing:

To test the many functionalities and pages of our site some automated tests were created using the testing tool selenium. With selenium we were able to create recordings of these tests where we show how the site works and how it responds to various interactions from the user. Some of these tests include but are not limited to correct and wrong credentials for sign in, missing required fields in forms, registering new clients and trainers, enrolling in classes etc.

Security Techniques/Measures:

The site uses sessions to make sure that all users that use our site have signed in. It forces people to sign in before gaining access to any other resources or pages of our site.

The site is using https protocol in order to make sure that all data coming or going in the connection between the user and the site is encrypted in case of man in the middle attacks.

The https protocol was also necessary in order to implement the physical sign in function and online payment.

PayPal requires certain scripts to ensure the safe transfer of money. These scripts and safety precautions along with certain certificates provided by PayPal were implemented with upmost attention and caution. All necessary measures have been taken to protect from dangers such as packet loss, fraudulent transactions, and identity theft.

For many actions even from within the site an email or password verification is required to make sure that the person performing these actions is the one that is signed in.

After registering a new user a random password is generated and given to the user, this password contains characters both upper and lower case and numbers. The user is then encouraged to change that password and provide something new that they can remember more easily if they want to.

Users can change their password if they wish. The password inserted by the user must include characters and numbers and its length must be equal or greater than eight characters.

When a user requests to change their password a verification key is generated and sent to their email. This key must be inserted in the change password procedure to make sure that the person requesting the password change is the actual user.

The user passwords are not known. Before storing the passwords in the database they are hashed. This is necessary so no one with access to the database has the ability to see and abuse the passwords and in case of a data breach/leak any information taken/stolen can't be used further.

Additional Notable Work

Navbar:

The navbar is a constant part of all pages that is used to help the user navigate the different parts of our page. The navbar must always be the same and unchanging so the user can find his/her way around the site even after visiting a page they have never seen before



Programs Announcements Memberships Nutritional Calculator 1RM Calculator Enrollment My Account Contact Us

Logout

(clients' navbar)



Home Announcements Statistics Subscriptions-Due Create Class Delete Class Client Data

Logout

(trainers' navbar)

Database Connection:

For the purposes of modularity, expandability, reusability and maintenance a file containing the code required for the connection to the database was created and is included in all the files that need access to the database.

Cronjobs:

For some functionalities some files needed to be executed in at certain times in specific intervals to ensure that our site is always up to date and correct. To achieve this some cronjobs were created to execute the necessary code when needed.

Physical Sign-In:

The gym wanted to know that clients that are signed into classes are attending, people that are attending classes are sign-in and not tricking the gym and finally for general statistics. To accomplish these goals a physical sign-in functionality was created to document the clients that enter the gym and at what times.

Client in the center of development

As part of our agile methodology, we worked with the client on the daily basis, showing them the application on many phases on development and getting constant feedback on the image and functionality of the application. Many changes were made because of this, including showcasing of certain statistics, class creation, memberships philosophy, sign in procedure and whether a user can see the applications contents without an account.

Web hosting

We setup a web hosting space with dedicated IP address, TLS certificate and https protocol in order to ensure safety, make the application public and available and finish both the physical sign in and payment functions of our application.