

MS Fit Care Gym

This is the official documentation of MS Fit Care Gym Website. Any developer can refer to this document for information about every website and every script used.

Features

- Dynamically edit,add,delete products,services,prices through ajax and php
- Image upload for every product and service
- Book Services Online
- User Registration and login services
- Edit,add,delete announcements
- Edit,delete users
- Different profiles (Admin,Physiotherapist,Customer,Massage Therapist)
- Edit General Program of Gym
- Edit personal program of customer
- Responsive, Bootstrap 4
- Sweetalert Implementation

MS Fit Care Gym is constructed using the free open source resources below:

Tech

Tech	README
Sweetalert	https://github.com/egoist/sweetalert/blob/master/README
Fontawesome	https://github.com/FortAwesome/Font-Awesome/blob/master/README.md
JQuery	https://github.com/jquery/jquery/blob/master/README.n
Bootstrap 4	https://github.com/tempusdominus/bootstrap-4/blob/master/README.md
Google Recaptcha	https://github.com/google/recaptcha

Shop

This section of the website is responsible to show to the customers, all the available products that are sold through the gym. The administrator profile has privileges to edit, add or delete products.

Features

- Dynamically edit,add,delete products through ajax and php
- Image upload for every product
- Sweetalert Implementation

Development

```
--> The files and dependancies of the shop reside in folder /shop
```

This function loads the bootstrap modal from add-product.html to add a product

shop.js

```
$('.add-btn-color').on('click', function(event){
    event.preventDefault();
    $('#modalAddProduct').modal('show').find('.modal-content').load($(this).attr('href'));
});
```

This function loads the products from the database into shop.php.

shop.js

```
function readProducts(){
    $('#load-products').load('read-products.php');
}
```

Fetching Products (read-products.php)

Below you can see the code that is fetching the products from the database. Firstly, a query is executed on mysql to bring the information from every single product. To maintain responsiveness, bootstrap is used and every 3 products, a new row is created. If an administrator profile is logged in, the option to add, delete or edit the products is

appeared next to each product. This website is loaded to /shop.php

```
<?php
    $sql = "SELECT * FROM products";
    $result = $DBcon->query($sql);
    $numOfCols = 3;
    $rowCount = 0;
    ?>
<div class="row padding-row">
    <?php
        while($row = $result->fetch(PDO::FETCH_ASSOC)){ ?>
            <div class="col-md-4 col-margin"><?php    if($_SESSION['
customer_id']>=1){
                if ($_SESSION['role']=='Admin') { ?>
                    <a class="edit-link" data-toggle="modal" data-targ
et="#modal
                        <?php echo $row["id"]; ?>">
                        <i class="fas fa-pencil-alt"></i>
                    </a>
                    <a class="edit-link" id="delete-product" data-id="
                        <?php echo $row["id"];?>" href="javascript:voi
d(0)">
                        <i class="fa fa-trash"></i>
                    </a>
                <?php    }
            } ?>
            <div class="card">
                " alt="">
        <div class="card-body">
            <h5 class="card-title">
                <?php echo $row["name"] ?>
            </h5>
            <p class="card-text">
                <?php echo $row["description"] ?>
            </p>
            <p class="card-text">
                <b>Price: </b> €
                <?php echo $row["price"] ?>
            </p>
        </div>
    </div>
</div>
<?php
$rowCount++;
if($rowCount % $numOfCols == 0) echo '
</div>
<div class="row padding-row">' ; ?>

```

Edit Product (shop.js,editPro.php)

For every product, a new bootstrap modal is constructed with a unique id, which is passed as val, to this javascript function. The data of the modal is passed to a FormData Object and with an ajax post request the

data is passed to a php webpage to execute a query to the database. When the data is successfully inserted into the database, the modal is hidden and the products section is reloaded.

shop.js

```
function update(val,event){
    var productID = val;

    var data = new FormData($("#edit-product"+val).
get(0));

    event.preventDefault();
    $.ajax({
        url: 'editPro.php',
        type: 'POST',
        data: data,
        contentType: false,
        processData: false,
        success: function(data) {

            $('#modal'+productID).modal('hide');
            $('body').removeClass('modal-open');
            $('.modal-backdrop').remove();
            readProducts();

        }
    })
}
```

```
        });  
    }  
}
```

editPro.php

When a post action is executed, we get the name of the file and set a location for the images to be uploaded (in this case the folder is uploads). The script accepts different types of images. The data from the previous website is saved to different variables and a query to the database is executed for the record to be inserted.

```
<?php  
require_once(' ../connect.php');  
$response = array();  
  
if (!empty($_POST)) {  
  
    $filename = $_FILES['file']['name'];  
  
  
    $location = '../uploads/'.$filename;  
  
  
    $file_extension = pathinfo($location, PATHINFO_EXTENSION);  
    $file_extension = strtolower($file_extension);  

```

```
$image_ext = array("jpg","png","jpeg","gif");

$response['img_status'] = 0;
if(in_array($file_extension,$image_ext)){

    if(move_uploaded_file($_FILES['file']['tmp_name'],$location)){

        $response['img_status'] = $location;
    }
}

$id = $_POST["id"];
$title = $_POST["ptitle"];
$description = $_POST["description"];
$price = $_POST["price"];


$query = "update products set name=:title, description=:description, img_file=:picture, price=:price where id=:id";

$stmt = $DBcon->prepare( $query );
$stmt->bindParam(':id', $id,PDO::PARAM_INT);
$stmt->bindParam(':title', $title);
$stmt->bindParam(':description', $description);
$stmt->bindParam(':picture', $filename);
$stmt->bindParam(':price', $price );
$stmt->execute();
```



```
        echo $json_encode($response);  
    }  
?>
```

Add Product (shop.js, insertPro.php)

shop.js

When the button with id #save is selected the data of the modal is passed to a FormData Object and with an ajax post request the data is passed to a php webpage to execute a query to the database. When the data is successfully inserted into the database, the modal is hidden and the products section is reloaded.

```
$(document).on('click', '#save', function(event) {  
  
    var data = new FormData($('#insert-product').get(0));  
  
    event.preventDefault();  
  
    $.ajax({  
        url: 'insertPro.php',  
        type: 'POST',  
        data: data,  
        contentType: false,  
        processData: false,  
        success: function(data) {
```

```

                                $('#modalAddProduct').modal('hide');
                                readProducts();
                            }
                        });
                    })

```

insertPro.php

When a post action is executed, we get the name of the file and set a location for the images to be uploaded (in this case the folder is uploads). The script accepts different types of images. The data from the previous website is saved to different variables and a query to the database is executed for the record to be inserted.

```

<?php

require_once('../connect.php');

if (!empty($_POST)) {
    $response = array();

    $filename = $_FILES['file']['name'];

    $location = '../uploads/'.$filename;

```

```
$file_extension = pathinfo($location, PATHINFO_EXTENSION);  
$file_extension = strtolower($file_extension);
```

```
$image_ext = array("jpg","png","jpeg","gif");
```

```
$response['img_status'] = 0;
```

```
if(in_array($file_extension,$image_ext)){
```

```
    if(move_uploaded_file($_FILES['file']['tmp_name'],$location)){
```

```
        $response['img_status'] = $location;
```

```
    }
```

```
}
```

```
    $title = $_POST["ptitle"];
```

```
    $description = $_POST["description"];
```

```
    $price = $_POST["price"];
```

```
    $query = "insert into products(name,description,image_file,price) values(:title, :description, :picture, :price)";
```

```
    $stmt = $DBcon->prepare( $query );
```

```
    $stmt->bindParam(':title', $title);
```

```
$stmt->bindParam(':description', $description);  
$stmt->bindParam(':picture', $filename);  
$stmt->bindParam(':price', $price);  
  
$stmt->execute();  
  
echo json_encode($response);  
  
}  
?>
```

Delete Product (shop.js,delete.php)

```
shop.js
```

The function below is created with the help of the plugin SweetAlert. The unique product id of the product is passed into the function. The SweetAlert let us change some properties of the prompt which are responsible for the design and animation of the box. After the user confirms the delete of the product, an ajax post request is generated and passes the product id to the delete.php file. Success and Fail messages are shown according to the response of the ajax request.

```
function SwalDelete(productId){

    swal({
        title: 'Are you sure?',
        text: "It will be deleted permanently!",
        type: 'warning',
        showCancelButton: true,
        confirmButtonColor: '#3085d6',
        cancelButtonColor: '#d33',
        confirmButtonText: 'Yes, delete it!',
        showLoaderOnConfirm: true,

        preConfirm: function() {
            return new Promise(function(resolve) {

                $.ajax({
                    url: 'delete.php',
                    type: 'POST',
                    data: 'delete='+productId,
                    dataType: 'json'
                })
                .done(function(response){
                    swal('Deleted!', response.message, response.status);

                    readProducts();
                })
                .fail(function(){
```

```

        swal('Oops...', 'Something went wrong
with ajax !', 'error');

    });

    },
    allowOutsideClick: false
});

}

```

This file executes a query to the database server and based on product ID deletes the record from the database. Also, a json response is generated and shown to the user.

delete.php

```

<?php

    header('Content-type: application/json; charset=UTF-8'
);

    $response = array();

    if ($_POST['delete']) {

        require_once '../connect.php';
    }

```

```

        $pid = intval($_POST['delete']);

        $query = "DELETE FROM products WHERE id=:pid";

        $stmt = $DBcon->prepare( $query );

        $stmt->execute(array(':pid'=>$pid));


        if ($stmt) {

            $response['status'] = 'success';

            $response['message'] = 'Product Deleted Successfully ...';

        } else {

            $response['status'] = 'error';

            $response['message'] = 'Unable to delete product ...';

        }

        echo json_encode($response);

    }

```

Prices

This section of the website is responsible to show to the customers, a pricelist of the gym. The administrator profile has privileges to edit, add or delete prices.

Features

- Dynamically edit,add,delete prices through ajax and php

- Sweetalert Implementation

Development

```
--> The files and dependancies of the prices reside in folder /prices
```

This function loads the bootstrap modal from add-prices.html to add a price

```
price.js
```

```
$('.add-btn-color').on('click', function(event){  
    event.preventDefault();  
    $('#modalAddPrice').modal('show').find('.modal-content').load($(this).attr('href'));  
});
```

This function loads the prices from the database into prices.php.

```
prices.js
```

```
function readPrices(){  
    $('#load-prices').load('read-prices.php');  
}
```


Fetching Prices (read-prices.php)

Below you can see the code that is fetching the prices from the database. Firstly, a query is executed on mysql to bring the information from every single price. If an administrator profile is logged in, the option to add,delete or edit the prices is appeared next to each price. This website is loaded to prices/prices.php

[illegible]

```

        <i class="fas fa-pencil-alt"></i>
    </a>
    <a class="edit-link" id="delete-product" data-id="
        <?php echo $row["id"];?>" href="javascript:void(0)">
        <i class="fa fa-trash"></i>
    </a>
</div>
<?php }
} ?>

<h5 class="mb-0">
    <button class="btn btn-link prices-titles"
        type="button" data-toggle="collapse" data-target="#collapse
        <?php echo $row["id"];?>" aria-expanded="true" aria-controls="collapse
        <?php echo $row["id"];?>">
        <h5>
            <i class="fas fa-plus"></i>
            <?php echo $row["title"]; ?>
        </h5>
    </button>
</h5>
</div>
<div id="collapse
    <?php echo $row["id"];?>" class="collapse" aria-labelledby="heading

```

```

        <?php echo $row["id"];?>" data-parent="#accordionExample">
        <div class="card-body">
            <h4>Price: €
                <?php echo $row["price"]; ?>
            </h4>
            <p>Description:
                <?php echo $row["description"]; ?>
            </p>
        </div>
    </div>
</div>

```

Edit Prices (prices.js,editPrice.php)

For every price item, a new bootstrap modal is constructed with a unique id, which is passed as val, to this javascript function. The data of the modal is serialized and with an ajax post request the data is passed to a php webpage to execute a query to the database. When the data is successfully inserted into the database, the modal is hidden and the prices section is reloaded.

```
prices.js
```

```
function update(val,event){
```

```
var priceID = val;

var data = $("#edit-price"+val).serialize();

event.preventDefault();
$.ajax({
    url: 'editPrice.php',
    type: 'POST',
    data: data,
    success: function(data) {

        $('#modal'+priceID).modal('hide');
        $('body').removeClass('modal-open');
        $('.modal-backdrop').remove();
        readPrices();

    }

});
}
```

editPrice.php

When a post action is executed, the data from the previous website is saved to different post variables and a query to the database is executed to insert the price record.

```
<?php
require_once('../connect.php');

$response = array();

if (!empty($_POST)) {

    $id = $_POST["id"];
    $title = $_POST["ptitle"];
    $description = $_POST["description"];
    $price = $_POST["price"];

    $query = "update prices set title=:title, price=:p
rice, description=:description where id=:id";
    $stmt = $DBcon->prepare( $query );
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);
    $stmt->bindParam(':title', $title);
    $stmt->bindParam(':price', $price);
    $stmt->bindParam(':description', $description);
    $stmt->execute();
    echo json_encode($response);
}

?>
```

Add Price (prices.js, prices.php)

prices.js

When the button with id #save is selected the data of the modal is passed to the insertPrice.php page with an ajax post request to execute a query to the database. When the data is successfully inserted into the database, the modal is hidden and the prices section is reloaded.

```
$(document).on('click', '#save', function(event) {  
    var data = $("#insert-prices").serialize();  
  
    event.preventDefault();  
    $.ajax({  
        url: 'insertPrice.php',  
        type: 'POST',  
        data: data,  
        success: function(data) {  
            $('#modalAddPrice').modal('hide');  
            readPrices();  
        }  
    });  
})
```

insertPrice.php

When a post action is executed, the data from the previous website is saved to different post variables and a query to the database is executed for the record to be inserted.

```
<?php
```

```
require_once('../connect.php');
```

```
if (!empty($_POST)) {
```

```
    $title = $_POST["ptitle"];
```

```
    $description = $_POST["description"];
```

```
    $price = $_POST["price"];
```

```
    $query = "insert into prices(title,price,description) values(:title, :price, :description)";
```

```
    $stmt = $DBcon->prepare( $query );
```

```
    $stmt->bindParam(':title', $title);
```

```
    $stmt->bindParam(':price', $price);
```

```
    $stmt->bindParam(':description', $description);
```

```
echo $json_encode($response);  
  
}  
?>
```

Delete Price (prices.js,delete.php)

prices.js

The function below is created with the help of the plugin SweetAlert. The unique price id of the price item is passed into the function. The SweetAlert let us change some properties of the prompt which are responsible for the design and animation of the box. After the user confirms the delete of the price, an ajax post request is generated and passes the price id to the delete.php file. Success and Fail messages are shown according to the response of the ajax request.

```
function SwalDelete(priceID){  
  
    swal({  
        title: 'Are you sure?',  
        text: "It will be deleted permanently!",  
        type: 'warning',  
        showCancelButton: true,  
        confirmButtonColor: '#3085d6',  
        cancelButtonColor: '#d33',  
        confirmButtonText: 'Yes, delete it!',  
        cancelButtonText: 'No, cancel!',  
        closeOnConfirm: true,  
        closeOnCancel: false,  
        showLoaderOnConfirm: true,  
        allowOutsideClick: false,  
        beforeConfirm: function() {  
            $.ajax({  
                url: 'delete.php',  
                type: 'POST',  
                data: {priceID: priceID},  
                success: function(response) {  
                    if(response === 'success') {  
                        SwalDelete(priceID);  
                    } else {  
                        SwalDelete(priceID);  
                    }  
                }  
            });  
        }  
    });  
}
```



```
confirmButtonText: 'Yes, delete it!',
showLoaderOnConfirm: true,

preConfirm: function() {
    return new Promise(function(resolve) {

        $.ajax({
            url: 'delete.php',
            type: 'POST',
            data: 'delete='+priceID,
            dataType: 'json'
        })
        .done(function(response){
            swal('Deleted!', response.message, response.status);

            readPrices();
        })
        .fail(function(){
            swal('Oops...', 'Something went wrong with ajax !', 'error');
        });
    });
},
allowOutsideClick: false
});

}
```

This file executes a query to the database server and based on price ID deletes the record from the database. Also, a json response is generated and shown to the user.

delete.php

```
<?php
    header('Content-type: application/json; charset=UTF-8'
);

$response = array();

if ($_POST['delete']) {

    require_once '../connect.php';

    $pid = intval($_POST['delete']);
    $query = "DELETE FROM prices WHERE id=:pid";
    $stmt = $DBcon->prepare( $query );
    $stmt->execute(array(':pid'=>$pid));

    if ($stmt) {
        $response['status'] = 'success';
        $response['message'] = 'Product Deleted Successfully ...';
    } else {
        $response['status'] = 'error';
    }
}
```

```
        $response['message'] = 'Unable to delete product ...';  
    }  
    echo json_encode($response);  
}
```

Services

This section of the website is responsible to show to the customers, all the available services that are offered through the gym. The administrator profile has privileges to edit, add or delete services.

Features

- Dynamically edit,add,delete services through ajax and php
- Image upload for every service
- Sweetalert Implementation

Development

```
--> The files and dependancies of the shop reside in folder /services
```

This function loads the bootstrap modal from add-services.html to add a product

```
services.js
```

```
$( '.add-btn-color' ).on( 'click', function( event ) {  
    event.preventDefault();  
    $( '#modalAddService' ).modal( 'show' ).find( '.modal-content' ).load( $( this ).attr( 'href' ) );  
});
```

This function loads the services from the database into services.php.

```
services.js
```

```
function readServices(){  
    $( '#load-service' ).load( 'read-service.php' );  
}
```

Fetching Services (read-service.php)

Below you can see the code that is fetching the services from the database. Firstly, a query is executed on mysql to bring the information from every single service. To maintain responsiveness, bootstrap is used and every 3 services, a new row is created. If an administrator profile is logged in, the option to add, delete or edit the services is appeared next to each service. This website is loaded to services/services.php

```

<?php

    $sql = "SELECT * FROM services";
    $result = $DBcon->query($sql);
    $numOfCols = 3;
    $rowCount = 0;

    ?>

<div class="row padding-row">
    <?php
        while($row = $result->fetch(PDO::FETCH_ASSOC)){ ?>
            <div class="col-md-4 col-margin"><?php if($_SESSION['
customer_id']>=1){
                if ($_SESSION['role']=='Admin') { ?>
                    <a class="edit-link" data-toggle="modal" data-targ
et="#modal
                        <?php echo $row["service_id"]; ?>">
                        <i class="fas fa-pencil-alt"></i>
                    </a>
                    <a class="edit-link" id="delete-product" data-id="
                        <?php echo $row["service_id"];?>" href="javasc
ript:void(0)">
                        <i class="fa fa-trash"></i>
                    </a>
                <?php }
            } ?>

            <div class="card">
                " alt="">

```

```

        <div class="card-body">
            <h5 class="card-title">
                <?php echo $row["title"] ?>
            </h5>
            <p class="card-text">
                <?php echo $row["description"] ?>
            </p>
            <p class="card-text">
                <b>Capacity: </b>
                <?php echo $row["capacity"] ?>
            </p>
        </div>
    </div>
</div>
<?php
$rowCount++;
if($rowCount % $numOfCols == 0) echo '
</div>
<div class="row padding-row">' ; ?>

```

Edit Service (services.js,editSer.php)

For every service, a new bootstrap modal is constructed with a unique id, which is passed as val, to this javascript function. The data of the modal is passed to a FormData Object and with an ajax post request the data is passed to a php webpage to execute a query to the database.

When the data is successfully inserted into the database, the modal is hidden and the services section is reloaded.

services.js

```
function update(val,event){
    var serviceID = val;
    var data = new FormData($("#edit-service"+val
    ).get(0));

    event.preventDefault();
    $.ajax({
        url: 'editSer.php',
        type: 'POST',
        data: data,
        contentType: false,
        processData: false,
        success: function(data) {

            $('#modal'+serviceID).modal('hide');
            $('body').removeClass('modal-open');
            $('.modal-backdrop').remove();
            readServices();

        }

    });
}
```

```
}
```

editSer.php

When a post action is executed, we get the name of the file and set a location for the images to be uploaded (in this case the folder is uploads). The script accepts different types of images. The data from the previous website is saved to different variables and a query to the database is executed for the record to be inserted.

```
<?php
require_once('../connect.php');

if (!empty($_POST)) {

    $filename = $_FILES['file']['name'];
    $location = '../uploads/'.$filename;
    $file_extension = pathinfo($location, PATHINFO_EXTENSION);
    $file_extension = strtolower($file_extension);
    $image_ext = array("jpg","png","jpeg","gif");

    $response['img_status'] = 0;
    if(in_array($file_extension,$image_ext)){

        if(move_uploaded_file($_FILES['file']['tmp_name'],$location)){

            $response['img_status'] = $location;
```



```

    }
}

$id = $_POST["id"];
$title = $_POST["title"];
$description = $_POST["description"];
$capacity = $_POST["capacity"];


$query = "update services set title=:title, description=:description, image=:image, capacity=:capacity where service_id=:id";

$stmt = $DBcon->prepare( $query );
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
$stmt->bindParam(':title', $title);
$stmt->bindParam(':description', $description);
$stmt->bindParam(':image', $filename);
$stmt->bindParam(':capacity', $capacity);


$stmt->execute();


echo json_encode($response);

}

?>

```

Add Service (services.js, insertSer.php)

services.js

When the button with id #save is selected the data of the modal is passed to a FormData Object and with an ajax post request the data is passed to a php webpage to execute a query to the database. When the data is successfully inserted into the database, the modal is hidden and the services section is reloaded.

```
$(document).on('click', '#save', function(event) {  
    var data = new FormData($("#insert-service").get(0));  
  
    event.preventDefault();  
    $.ajax({  
        url: 'insertSer.php',  
        type: 'POST',  
        data: data,  
        contentType: false,  
        processData: false,  
        success: function(data) {  
            $('#modalAddService').modal('hide');  
            readServices();  
        }  
    });  
})
```

```
insertSer.php
```

When a post action is executed, we get the name of the file and set a location for the images to be uploaded (in this case the folder is uploads). The script accepts different types of images. The data from the previous website is saved to different variables and a query to the database is executed for the record to be inserted.

```
<?php
require_once('../connect.php');

if (!empty($_POST)) {
$response = array();
$filename = $_FILES['file']['name'];
$location = '../uploads/'.$filename;
$file_extension = pathinfo($location, PATHINFO_EXTENSION);
$file_extension = strtolower($file_extension);
$image_ext = array("jpg","png","jpeg","gif");
$response['img_status'] = 0;
if(in_array($file_extension,$image_ext)){

    if(move_uploaded_file($_FILES['file']['tmp_name'],$location)){
        $response['img_status'] = $location;
    }
}
```

```

}

$title = $_POST["title"];
$description = $_POST["description"];
$capacity = $_POST["capacity"];


$query = "insert into services(title,description,c
apacity,image) values(:title, :description, :capacity, :im
age)";

$stmt = $DBcon->prepare( $query );

$stmt->bindParam(':title', $title);
$stmt->bindParam(':description', $description);
$stmt->bindParam(':capacity', $capacity);
$stmt->bindParam(':image', $filename);


$stmt->execute();
echo json_encode($response);
}
?>

```

Delete Service (services.js,delete.php)

```
services.js
```

The function below is created with the help of the plugin SweetAlert.

The unique service id of the service is passed into the function. The SweetAlert let us change some properties of the prompt which are responsible for the design and animation of the box. After the user confirms the delete of the service, an ajax post request is generated and passes the service id to the delete.php file. Success and Fail messages are shown according to the response of the ajax request.

```
function SwalDelete(serviceID){  
    swal({  
        title: 'Are you sure?',  
        text: "It will be deleted permanently!",  
        type: 'warning',  
        showCancelButton: true,  
        confirmButtonColor: '#3085d6',  
        cancelButtonColor: '#d33',  
        confirmButtonText: 'Yes, delete it!',  
        showLoaderOnConfirm: true,  
  
        preConfirm: function() {  
            return new Promise(function(resolve) {  
  
                $.ajax({  
                    url: 'delete.php',  
                    type: 'POST',  
                    data: 'delete='+serviceID,  
                    dataType: 'json'  
                })  
  
                .done(function(response){
```

```

        swal('Deleted!', response.message, response.status);

        readServices();
    })
    .fail(function(){
        swal('Oops...', 'Something went wrong with ajax !', 'error');
    });
});
},
allowOutsideClick: false
});
}

```

This file executes a query to the database server and based on service ID deletes the record from the database. Also, a json response is generated and shown to the user.

```
delete.php
```

```

<?php
    header('Content-type: application/json; charset=UTF-8');

    $response = array();

```

```

if ($_POST['delete']) {

    require_once '../connect.php';

    $pid = intval($_POST['delete']);
    $query = "DELETE FROM services WHERE service_id=:p
id";

    $stmt = $DBcon->prepare( $query );
    $stmt->execute(array(':pid'=>$pid));

    if ($stmt) {
        $response['status'] = 'success';
        $response['message'] = 'Service Deleted Succes
sfully ...';
    } else {
        $response['status'] = 'error';
        $response['message'] = 'Unable to delete servi
ce ...';
    }

    echo json_encode($response);
}

```

Announcements

We take information from database and specialy from the table announcements and shows to the users of the website the announcements of the gym . There are buttons to redirect the only the

users whose role is Administrator, in another files for add , delete , or edit announcements. A visitor of the website who is not logged in can see the announcements .

Features!

- adding announcements
- editing announcements
- removing announcements
- update database according to the request given

Development

The files and dependancies of the announcements reside in folder /announcements

Annoucements.php

Only the users with role administrator and while they are logged in can see the button <Add Announcement> which redirect them in the file createAnn.php

```
<?php  if($_SESSION['customer_id']>=1){  
    if ($_SESSION['role']=='Admin') { ?>  
    <a class="btn btn-primary btn-lg round color-ann-btn" href=  
f="createAnn.php" role="button"> Add Announcement </a>  
    <?php  }
```



```
} ?>  
<br>
```

We select from the database all the records of the table announcements and print them. For every announcement there is the option to delete or edit . (Only for Administrator users .)

```
<?php $query = mysqli_query($conn,"select * from announcements");  
    while ($table=mysqli_fetch_array($query)) { ?>  
  
        <h2 class="ann-id"> Announcement <?php echo $table["id"] ?></h2>  
  
        <i class="fas fa-calendar-alt fa-5x"></i>  
        <h2 class="ann-id py-3"> Date: <?php echo $table["date"] ?></h2>  
  
        <?php if($_SESSION['customer_id']>=1){  
            if ($_SESSION['role']=='Admin') { ?>  
                <div class="admin-panel">  
                    <a href="EditA.php?id=<?php echo $table["id"];?>"><i class="fas fa-pencil-alt"></i></a>  
                    <a href="delete.php?id=<?php echo $table["id"];?>"><i class="fas fa-trash"></i></a>  
                </div>  
                <?php }  
            } ?>  
        <h4 class="ann-heading">Subject: <?php echo
```

```
$table["subject"] ?></h4>
```

```
<p class="ann-text"><?php echo $table["maint  
ext"] ?> </p>
```

```
<?php } ?>
```

createAnn.php

This file is responsible for creating new announcements, the Admin user complete the date, subject, and content of the announcement.

```
<h5 class="card-title text-center">Create a new Announceme  
nt</h5>
```

```
<?php
```

```
if ($flag ==1) {
```

```
    echo '<div class="alert alert-success"><st  
rong>The announcement was posted successfully. </strong>  
</div>';
```

```
}
```

```
?>
```

```
<form action="createAnn.php" method="post" cla  
ss="form-signin">
```

```
<input type="date" class="form-control" na  
me="date" placeholder="Select the date of the announcement  
" />
```

```
<input type="text" class="form-control" na  
me="subject" placeholder="What is the subject?"/>
```

```
        <input type="text" class="form-control" name="maintext" placeholder="Add the new content"/>

        <input type="submit" class="btn btn-lg btn-primary btn-block text-uppercase" name="send" type="submit" value="Add"/>

    </form>
```

When the user clicks on the button submit, then the information is registered as a new record in the table announcements of the gym database.

```
$flag=0;

mysqli_select_db($conn,"gym");
if(isset($_POST['send'])) {
    $date = $_POST['date'] ;
    $subject = $_POST['subject'] ;
    $message = $_POST['maintext'] ;

    $query = "insert into announcements (date,subject,main
text) values('$date','$subject','$message')";
    $sqlcrt=mysqli_query($conn,$query);
    $flag=1;
```

delete.php

When the button is pressed, the transition to the delete.php file also transfers the ID of the specific announcement so it can be deleted from the base. Takes the announcement ID that is unique to each record on the table, identifies it and deletes it.

```
$id = $_GET['id'];  
mysqli_select_db($conn,"database");  
$sqldel=mysqli_query($conn,"delete from announcements wh  
ere id='$id'");  
header('Location: announcement.php');
```

EditA.php

It takes the number of the announcement, which is unique to each record in the table, finds it, loads its data from the database, so the user can see the information of that announcement and change them. If any element is not changed it remains the same on the database.

```
<?php $id = $_GET['id'];  
$fetch_ann = mysqli_query($conn,"select * from announc  
ements where id='$id'");  
$fetched_ann=mysqli_fetch_array($fetch_ann); ?>  
<h5 class="card-title text-center">Edit an Ann  
ouncement</h5>  
<?php  
if ($flag ==1) {  
    echo '<div class="alert alert-success"><st
```

```

rong>The announcement was updated successfully. </strong>
</div>';

    }

    ?>

    <form action="createAnn.php" method="post" cla
ss="form-signin">

        <label for="date">Date</label>
        <input type="date" class="form-control" na
me="date" value="<?php echo $fetchd_ann["date"] ?>" />

        <label for="date">Subject</label>
        <input type="text" class="form-control" na
me="subject" value="<?php echo $fetchd_ann["subject"] ?>"
/>

        <input type="text" class="form-control" na
me="maintext" value="<?php echo $fetchd_ann["maintext"] ?
>" />

        <input type="submit" class="btn btn-lg btn-p
rimary btn-block text-uppercase" name="send" type="submit"
value="Add"/>
    </form>

```

When the submit button is pressed it refreshes the changed elements in the database.

```
mysqli_select_db($conn,"gym");
if(isset($_POST['send'])) {
    $date = $_POST['date'] ;
    $subject = $_POST['subject'] ;
    $message = $_POST['maintext'] ;

    $sqledit=mysqli_query($conn,"UPDATE announcements SET
date='$date', subject='$subject', maintext='$message' where
id='$id'");
    $flag=1;
}
```

Program

The program is loaded from the database is in an image format. It is possible to change the program only if it is in an image format, through the edit button that takes the user to another file to change the program.

Features!

- View the general program of the gym
- Upload/Change the program of the gym

Development

The files and dependancies of the program reside in folder
/programs

program.php

Loads the image that is present in the first record of the table program .
Only the users with role administrator and while they are logged in can
see the button <Edit Program> which redirect them in the file
programE.php

```
<?php
$id=1;
$query = mysqli_query($conn,"select pdf from program
where id='$id'");
while ($row = mysqli_fetch_assoc($query)) {
$pdf=$row['pdf'];
} ?>

<?php if($_SESSION['customer_id']>=1){
    if ($_SESSION['role']=='Admin') { ?>
        <div class="px-3 py-3">
            <a class="btn btn-primary btn-lg round color-ann-btn " href="programE.php" role="button"> Edit Program
        </a>
    <?php }
    } ?>
```

programE.php

The Administrator user can change the program by uploading a new image .

```
<form action="programE.php" method="post" enctype="multipart/form-data" class="form-signin">
    <label for="file">Select File</label>
    <input type="file" class="form-control" name="file" />
    <input type="submit" class="btn btn-lg btn-primary btn-block text-uppercase" name="send" type="submit" value="Add"/>
</form>
```

When the submit button is pressed, it checks that the file format is an image and sends a message to the user, if it is, then it updates the program.

```
$message="";
$id=1;
if (isset($_POST['send'])) {
    if (isset($_FILES["file"]["name"])) {

        $name = $_FILES["file"]["name"];
        $tmp_name = $_FILES['file']['tmp_name'];
```



```

$error = $_FILES['file']['error'];
$parts = explode(".", $name);
$one = $parts[0];
$two = $parts[1];
if (!empty($name)&&($two=='jpg' || $two=='gif' || $two=='j
peg' || $two=='heic')) {
    $location = '../uploads/';

    if (move_uploaded_file($tmp_name, $location.$name
)){
        $sqledit=mysqli_query($conn,"UPDATE program SET pdf='
$location$name' where id='$id'");
        $message = 'Uploaded';
    }

} else {
    $message= 'Please select only images!';
}
}
}
?>

```

Registration

Every visitor to the Gym website has the right to create an account on the website. In order for the account to be created, the user must complete the necessary information ,which are the email and username

that are unique to each user, the name , the password that must contain a capital letter, a lowercase letter, and a number. The user must agree with the terms of the site, and declare that it is not a robot for safety reasons .

Features!

- Create a user account

Development

The files and dependancies of the program reside in folder
/registration

register.php

The form that the user will fill in to create an account on the website.

```
<form action="register.php" method="post"
    <input type="text" id="inputEmail" class="
form-control" name="name" placeholder="*Name" />
    <input type="text" class="form-control" na
me="surname" placeholder="*Surname" />
    <input type="number" id="inputEmail" class
="form-control" name="telephone" placeholder="Telephone" /
>
    <input type="text" id="inputEmail" class="
```

```

form-control" name="email" placeholder="*Email address" />
        <input type="text" class="form-control" na
me="username" placeholder="*Username" />
        <input type="password" id="inputPassword"
class="form-control" name="password" placeholder="Password
"/>
        <input type="radio" class="form-check-input" v
alue="Male" name="sex">Male
        </label>
        <input type="radio" class="form-check-input" v
alue="Female" name="sex">Female
        </label>
        <input type="radio" class="form-check-input" v
alue="Other" name="sex">Other
        <input type="checkbox" name = "tos" class="
form-check-input" id="exampleCheck1">
        <label class="form-check-label" for="examp
leCheck1">By checking this box you agree to the <a class="
tos-terms" data-toggle="modal" data-target="#tosModal">Ter
ms and Conditions of our website.</a></label>
        <input type="submit" class="btn btn-lg btn-reg
ister btn-block text-uppercase" name="btnRegister" value="
Register"/>
    </form>

```

The information completed by the user are checked for their validity. For example, the email and username must be unique, the code must

comply with the rules, the user must comply with the terms of the page

```
if ($_POST['name'] == "") {
    $register_error_message = 'Name field is required!';
}
} else if ($_POST['surname'] == "") {
    $register_error_message = 'Surname field is required!';
} else if ($_POST['tos'] == "") {
    $register_error_message = 'You did not agree to the terms and conditions';
}
else if ($_POST['email'] == "") {
    $register_error_message = 'Email field is required!';
} else if ($_POST['username'] == "") {
    $register_error_message = 'Username field is required!';
} else if ($_POST['password'] == "") {
    $register_error_message = 'Password field is required!';
} else if (preg_match("/^.*(?=.{8,})(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z]).*$/", $_POST['password']) === 0){
    $register_error_message = '<p class="errText">Password must be at least 8 characters and must contain at least one lower case letter, one upper case letter and one digit</p>';
} else if (!filter_var($_POST['email'], FILTER_VALIDATE
```

```

_EMAIL)) {
    $register_error_message = 'Invalid email address!';
;
    } else if ($app->isEmail($DBcon,$_POST['email'])) {
        $register_error_message = 'Email is already in use!';
    } else if ($app->isUsername($DBcon,$_POST['username'])) {
        $register_error_message = 'Username is already in use!';
    } else {

        $customer_id=$app->Register($DBcon,$_POST['name'],$_POST['surname'],$_POST['telephone'],$_POST['email'], $_POST['username'], $_POST['password'], $_POST['sex'] );

        $_SESSION['customer_id'] = $customer_id;

        header("Location: register-success.php");
    }
}

```

regChecker.php

Some of the checks of the data is done by calling functions found in the file regChecker.php. It checks if the user's email or username is on the database does not allow user to use them.

```
public function isUsername($DBcon,$username)
{
    try {
        //$db = new PDO("mysql:host=$servername;dbname
=$database", $username, $password);
        $query = $DBcon->prepare("SELECT customer_id F
ROM customers WHERE username=:username");
        $query->bindParam("username", $username, PDO::
PARAM_STR);
        $query->execute();
        if ($query->rowCount() > 0) {
            return true;
        } else {
            return false;
        }
    } catch (PDOException $e) {
        exit($e->getMessage());
    }
}

public function isEmail($DBcon,$email)
{
    try {
        // $db = new PDO("mysql:host=$servername;dbname
=$database", $username, $password);
        $query = $DBcon->prepare("SELECT customer_id F
ROM customers WHERE email=:email");
        $query->bindParam("email", $email, PDO::PARAM_
```

```

STR);

$query->execute();

if ($query->rowCount() > 0) {
    return true;
} else {
    return false; }

} catch (PDOException $e) {
    exit($e->getMessage());
} }

```

The correct information we get from the user, we send them to the function which makes the registration on the database

```

public function Register($DBcon,$name, $surname, $telephone
, $email, $username, $password , $sex)
{
    try {
        //$db = new PDO("mysql:host=$servername;dbname
=$database", $username, $password);
        $query = $DBcon->prepare("INSERT INTO customer
s(name, surname , telephone , email, username, password ,
sex) VALUES (:name, :surname,:telephone,:email,:username,:
password ,:sex)");
        $query->bindParam("name", $name, PDO::PARAM_ST
R);
        $query->bindParam("surname", $surname, PDO::PA
RAM_STR);

```

```

        $query->bindParam("telephone", $telephone, PDO::PARAM_STR);

        $query->bindParam("email", $email, PDO::PARAM_STR);

        $query->bindParam("username", $username, PDO::PARAM_STR);

        $enc_password = hash('sha256', $password);
        $query->bindParam("password", $enc_password, PDO::PARAM_STR);

        $query->bindParam("sex", $sex, PDO::PARAM_STR);

        $query->execute();


        return $DBcon->lastInsertId();
    } catch (PDOException $e) {
        exit($e->getMessage());
    }
}

```

Logout

Destroy the session and logs out the users

Features!

- Destroy user's session

Development

The files and dependancies of the program reside **in** folder
/registration

logout.php

```
<?php

session_start();

// Destroy user session
unset($_SESSION['customer_id']);

// Redirect to index.php page
header("Location: login.php");

?>
```

Login

Login is a php file that represents the login page of a website. It displays the needed credentials that have to be given correctly in order to login the website. It also offers the option to create an account (register) and the option to (forgot my password) if the user has forgotten his password.

Features!

- login the website
- create an account
- forgot my password form

```
--> The files and dependancies of the Login reside in folder /registration
```

Code

Login Form

Checks if the btnlogin was requested if not the check if the parameters are filled. Until the user fills in the parameters needed print that the parameter which is empty is needed to login. If the parameteres are filled then call the login function which is located in the loginChecker php file.

When the loginChecker checks if the username and password match to a user then fill the \$_SESSION with all its information so they are accesible from all the pages.

```
if (!empty($_POST['btnLogin'])) {  
  
    $username = trim($_POST['username']);  
    $password = trim($_POST['password']);
```

```
if ($username == "") {  
    $login_error_message = 'Username field is required  
!';  
} else if ($password == "") {  
    $login_error_message = 'Password field is required  
!';  
} else {  
    $customer_id = $app->Login($DBcon,$username, $pass  
word);  
    if($customer_id > 0)  
    {  
        $_SESSION['customer_id'] = $customer_id;  
        $query = $DBcon->prepare("SELECT customer_id,  
name, username, email , role , password , surname , teleph  
one , program FROM customers WHERE customer_id=:customer_i  
d");  
        $query->bindParam("customer_id", $customer_id,  
PDO::PARAM_STR);  
        $query->execute();  
        $user=$query->fetch(PDO::FETCH_OBJ);  
        $_SESSION['role'] = $user->role;  
        $_SESSION['surname']=$user->surname;  
        $_SESSION['name']= $user->name;  
        $_SESSION['email']= $user->email;  
        $_SESSION['telephone']= $user->telephone;  
        $_SESSION['username']= $user->username;  
        $enc_password = hash('sha256', $password);  
        $_SESSION['password']=$enc_password;
```

```
$_SESSION['sex'] = $user->sex;
$_SESSION['program'] = $user->program;
header("Location: ../index.php");
}
```

Login Checker

The loginChecker gets the 2 paramters needed from the login.php file and checks if their is a username that matches with the password given in the database (customers). Before trying to match the password given with the password that is stored in the the database we encrypt the passsword given with sha256 because all the password are encrypted and then entered in the database when created.

If they match then return to the login php file which will redirect the user to the index php file (Home page).

```
public function Login($DBcon, $username, $password)
{
    try {

        $query = $DBcon->prepare("SELECT customer_id F
ROM customers WHERE (username=:username OR email=:username
) AND password=:password");

        $query->bindParam("username", $username, PDO::
PARAM_STR);

        $enc_password = hash('sha256', $password);

        $query->bindParam("password", $enc_password, P
```

```

PDO::PARAM_STR);

    $query->execute();

    if ($query->rowCount() > 0) {

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

        return $result->customer_id;

    } else {

        return false;

    }

} catch (PDOException $e) {

    exit($e->getMessage());

}

}

```

Profile

Profile is a php file that presents to each user it's personal information (name, ,surname, username etc), personal booking that she/he has booked and also it offers an option to check the personal gym program that the users has. The Profile file hides or shows different content based on the role of the user. The page also offers the capability to the user with role (user) to change their personal information and for users with role (admin) to edit, add, delete information

```

--> The files and dependancies of the Profile reside in folder /profile

```

Features!

- See personal information
- See personal program
- See personal booking
- Edit personal information
- Able to edit users (if role == admin)
- Able to delete users (if role == admin)
- See all the bookings closed (if role == admin)

The files and dependancies of the Profile reside in folder /profile

Code

Checks if there is a user logged in the page, if so then it check if the users role is an admin. If all true then show a button (Users Management) that redirects to the seeUsers php file.

```
<?php if($_SESSION['customer_id']>=1){  
    if ($_SESSION['role']=='Admin') { ?>  
        <a href="seeUsers.php" class="btn btn-lg btn-success  
round float-right">Users Management</a>  
        <?php }  
    } ?>
```

Personal Information

It uses the customer id that is saved in \$_SESSION to fetch all the users data from the database (customers) by using mysqli commands, and then it present them to the user. The containt can be changed and by pressing the submit button it will update the database with the new information of the user.

```
<?php $id = $_SESSION['customer_id'];  
    $fetch_ann = mysqli_query($conn,"select * from customers where customer_id='$id'");  
    $fetched_ann=mysqli_fetch_array($fetch_ann); ?>  
  
    <div class="card card-signin">  
        <div class="card-body">  
            <h5 class="card-title text-center manage-title">Manage Your Profile</h5>  
            <form class="form-signin" action="profile.php?id=<?php echo $id;?>" method="post">  
                <div class="form-label-group">  
                    <label for="name">Name</label>  
                    <input type="text" id="name" name="name" class="form-control" value="<?php echo $fetched_ann["name"] ?>">  
                </div>  
                <div class="form-label-group">  
                    <label for="name">Surname</label>  
                    <input type="text" id="surname" name="surname" class="
```

```
form-control" value="<?php echo $fetchd_ann["surname"] ?
>">

</div>

<div class="form-label-group">
<label for="Email">Email</label>
<input type="email" id="email" name="email" class="fo
rm-control" value="<?php echo $fetchd_ann["email"] ?>">
</div>

<div class="form-label-group">
<label for="Username">Username</label>
<input type="text" id="username" name="username" class
="form-control" value="<?php echo $fetchd_ann["username"]
?>">
</div>

<div class="form-label-group">
<label for="Telephone">Telephone</label>
<input type="number" id="telephone" name="telephone" c
lass="form-control" value="<?php echo $fetchd_ann["teleph
one"] ?>">
</div>

<input type="submit" class="btn btn-lg btn-primar
y btn-block text-uppercase" name="send" type="submit" valu
e="Change info"/>
```

Check for valid information

Check if the information changed by the user are okay to be used and uploaded to the database when pressing the submit button.

Checks if the email, username are already in use and it also checks that the information given is valid and supported by the systems requirements. If not then the appropriate message will appear to inform the user what is wrong.

```
mysqli_select_db($conn,"database");

if(isset($_POST['send'])) {

    $name = $_POST['name'] ;

    $surname = $_POST['surname'] ;

    $email = $_POST['email'] ;

    $username = $_POST['username'] ;

    $telephone = $_POST['telephone'] ;


    $query1 = mysqli_query($conn,"SELECT * FROM custom
ers WHERE email='$email'");

    $num_rows1 = mysqli_num_rows($query1);

    $query2= mysqli_query($conn,"SELECT * FROM custome
rs WHERE username='$username'");

    $num_rows2 = mysqli_num_rows($query2);


    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "Invalid email address!";
    }else if((strcmp($email,$_SESSION['email'])!=0)&&$
num_rows1>0){

        $query1 = mysqli_query($conn,"SELECT customer_
```

```

id FROM customers WHERE email='$email'");
    while ($row = mysqli_fetch_assoc($query1))
        $numid=$row['customer_id'];
    echo "hey" . $numid;
    echo "hi" . $id;
    if($numid!=$id){
        echo "Email " . $email . " is already in use!";
    }else{
        $sqledit=mysqli_query($conn,"UPDATE customers
SET name='$name' , surname='$surname' , email='$email' , u
sername='$username' where customer_id='$id'");
        $query1 = mysqli_query($conn,"SELECT surname ,
name , email , telephone , username FROM customers WHERE
customer_id='$id'");
        while($row =mysqli_fetch_assoc($query1)) {
            $_SESSION['surname']=$row["surname"];
            $_SESSION['name']= $row["name"];
            $_SESSION['email']= $row["email"];
            $_SESSION['telephone']=$row["telephone"];
            $_SESSION['username']=$row["username"];
        }
        header('Location: profile.php');
    }
}
}else if((strcmp($username,$_SESSION['username'])
!=0)&&$num_rows2>0){
    $query1 = mysqli_query($conn,"SELECT customer_id F
ROM customers WHERE username='$username'");
    while ($row = mysqli_fetch_assoc($query1))

```

```

        $numid=$row['customer_id'];

        if($numid!=$id){
            echo "Username " . $username . " is already in use
!";
        }else{
            $sqledit=mysqli_query($conn,"UPDATE customers
SET name='$name' , surname='$surname' , email='$email' , u
sername='$username' where customer_id='$id'");
            $query1 = mysqli_query($conn,"SELECT surname ,
name , email , telephone , username FROM customers WHERE
customer_id='$id'");
            while($row =mysqli_fetch_assoc($query1)) {
                $_SESSION['surname']=$row["surname"];
                $_SESSION['name']= $row["name"];
                $_SESSION['email']= $row["email"];
                $_SESSION['telephone']=$row["telephone"];
                $_SESSION['username']=$row["username"];
            }
            echo '<meta http-equiv="refresh" content="0">';
        }

    }else{

        $sqledit=mysqli_query($conn,"UPDATE customers SET name=
'$name' , surname='$surname' , email='$email' , username=
'$username' , telephone='$telephone' where customer_id='$id
'");
    }

```

```

        $query1 = mysqli_query($conn,"SELECT surname ,
name , email , telephone , username FROM customers WHERE
customer_id='$id'");

        while($row =mysqli_fetch_assoc($query1)) {
            $_SESSION['surname']=$row["surname"];
            $_SESSION['name']= $row["name"];
            $_SESSION['email']= $row["email"];
            $_SESSION['telephone']=$row["telephone"];
            $_SESSION['username']=$row["username"];
        }

        echo ' <meta http-equiv="refresh" content="0">' ;} } ?>

```

Personal program

Checks if there is a program in the \$_SESSION. If true then print the button (Download Program) that when it gets pressed it show the users program

If the user is an admin then dont show the button.

```

<?php    if(!empty($_SESSION['program'])){ ?>
        <a href="<?php echo $_SESSION['program'] ?>" class="btn btn-lg btn-block btn-danger text-uppercase ">Download Program</a>

        $user='Admin';

        if(strcmp($user,$_SESSION['role'])!=0){ ?>

```

```
<a class="btn btn-lg btn-block btn-danger text
-uppercease disabled">Download Program</a>

<?php  }

} ?>
```

Reset Password

It presents a button (Reset Password) that redirects the user to the resetPass.php file that allows the users to change its password by following the steps that they will appear.

```
<a href="resetPass.php" class="btn btn-lg btn-success btn-
block text-uppercease">Reset Password</a>
```

Show Bookings

Finds the role and the username of the user from the \$_SESSION and it prints all the bookings he has made by using msqli commands. If there are not any then print NO BOOKINGS.

```
<div class="profile-section mx-auto">
    <div class="container ">
        <h1 class="text-center py-3" id="book-h1"> Boo
kings </h1>
        <p class="text-center" id="book-p"> You can se
e below, all the bookings you have made throught our websi
te.</p>
```

```

        <br>

    <?php
        $role=$_SESSION['role'];
        $user=$_SESSION['username'];

        $sql9 = "SELECT COUNT(*) from book WHERE username='$user'
        ";

        $use = mysqli_query($conn, $sql9);
        $row9 = mysqli_fetch_array($use);
        $userrows=$row9[0];

        $sql10 = "SELECT * FROM book WHERE username='$user'";

        $result10 = mysqli_query($conn, $sql10);

        $c=0;
        if ($userrows > 0) {

            while ($row10 = mysqli_fetch_assoc($result10)) { ?
>
                <div class="row bookings">

                    <div class="col-md-3">
                        <p >Booking ID: <?php echo $row10["id"]; ?
></p>

                    </div>

                    <div class="col-md-3">

```

```

        <p> Service: <?php echo $row10["service"]; ?>
    </p>
    </div>
    <div class="col-md-3">
        <p>Date: <?php echo date('m/d/Y',$row10["day"]);?></p>
    </div>
    <div class="col-md-3">
        <p>Time: <?php echo $row10["time"]; ?></p>
    </div>

    </div>
    <br>
    <?php } } else {
        echo '<h1> No bookings to show</h1>';
    }

?>

```

The redirected files and their factuality are:

```
seeUsers.php
```

The seeUsers php file presents to the admin all the users and their information.

It gives the admin the ability to change any users role and the program. It can also delete a user by prassing the button ()

```
while ($row=mysqli_fetch_array($sql)) { ?>
    <tr>
        <td><p ><?php echo $row["name"]; ?></p></td>
        <td><p><?php echo $row["surname"]; ?></p></td>
        <td> <p> <?php echo $row["telephone"]; ?></p></td>
        <td> <p> <?php echo $row["email"]; ?></p></td>
        <td> <p><?php echo $row["username"]; ?></p></td>
        <td> <p><?php echo $row["sex"]; ?></p></td>
        <td><p><?php echo $row["role"]; ?></p></td>
        <td><?php echo ' <a href="editU.php?customer_id='.$row["customer_id"].' "> <i class="fas fa-pencil-alt"></i></a>
```



```

        <a href="deleteU.php?customer_id='.$row["customer_
id"].'"> <i class="fa fa-trash"></i> </a>'; ?> </td>

    </tr>

```

editU.php

The editU.php file checks what changes have been made to the user information and prints a message that it represents the changes that have been made and updates the database with the new information of the user.

```

$flag_msg="";

$id = $_GET['customer_id'];
mysqli_select_db($conn,"database");

if(isset($_POST['send'])) {

$name = $_FILES['file']['name'];
$temp_name = $_FILES['file']['tmp_name'];
if(isset($name)){
    if(!empty($name)){
        $location = '../uploads/';
        if(move_uploaded_file($temp_name, $location.$n
ame)){
            if(empty($_POST['role'])){
                $sqledit=mysqli_query($conn,"UPDATE customers SET pro
gram='$location$name' where customer_id='$id'");
                $flag_msg='Only the program of the use

```

```
r has been updated successfully' ;  
} else {  
    $role = $_POST['role'] ;  
    $sqledit=mysqli_query($conn,"UPDATE customers SET role='  
$role' , program='$location.$name' where customer_id='$id'  
");  
    $flag_msg='The role and the program of the  
user have been updated successfully' ; }  
}  
}else{  
    if(empty($_POST['role'])){  
        $flag_msg='No any updates' ;  
    } else{  
        $role = $_POST['role'] ;  
        $sqledit=mysqli_query($conn,"UPDATE customers SET  
role='$role' where customer_id='$id'");  
        $flag_msg='Only the role of the user has been updat  
ed successfully' ;  
    }  
}  
} else {  
    if(empty($_POST['role'])){  
        $flag_msg='No any updates' ;  
    } else{  
        $role = $_POST['role'] ;  
        $sqledit=mysqli_query($conn,"UPDATE customers SET
```

```

role='$role' where customer_id='$id'");
    $flag_msg='Only the role of the user has been updated successfully' ;
}
}

```

deleteU.php

The deleteU php file deletes the user from the database with the id that was selected from the admin by using mysqli commands

```

$id = $_GET['customer_id'];

mysqli_select_db($conn,"database");
$sqldel=mysqli_query($conn,"delete from customers where customer_id='$id'");
header("Location: seeUsers.php")
?>

```

resetPass.php

The resetPass php file is responsible to allow the users to change their passwords by filling the form. The form needs them to give their old password 1 time and the new password that they want 2 times. Then a series of checks run to check that the password given is correct and that the new password requested is valid and it matches the 2nd new

password given.

```
if (!empty($_POST['btnLogin'])) {

    $currentpassword = trim($_POST['currentpassword']);
    $NEWpassword = trim($_POST['NEWpassword']);
    $RnewPassword= trim($_POST['RnewPassword']);
    $enc_password = hash('sha256', $currentpassword);

    if ($currentpassword == "") {
        $login_error_message = 'Current password field is
required!';
    } else if ($NEWpassword == "") {
        $login_error_message = 'New Password field is requ
ired!';
    }else if ($RnewPassword==""){
        $login_error_message = 'Rewrite new Password field
is required!';

    }else if((strcmp($enc_password,$_SESSION['password'])!=
=0)){
        $login_error_message = 'Wrong current password';

    }else if((strcmp( $NEWpassword,$RnewPassword)!=0)){
        $login_error_message = 'The two fields of new pass
word are not the same';

    }else if(preg_match("/^.*(?:=.{8,})(?=.*[0-9])(?=.*[a-z
```

```

    ))(?!.*[A-Z]).*$/" , $_POST['NEWpassword']) === 0){
        $login_error_message = '<p class="errText">Passwor
d must be at least 8 characters and must contain at least
one lower case letter, one upper case letter and one digit
</p>';
    } else {
        $enc_passwordF = hash('sha256', $NEWpassword);

        $id=$_SESSION['customer_id'];
        $sqledit=mysqli_query($conn," UPDATE customers SET passw
ord='$enc_passwordF' where customer_id= '$id'");
        $flag="B";
    }
}
?>

```

Forgot Password

This is the part of the website that is responsible for giving a user the ability to reset his password in case he forgot it. The user types his/her username and an automated email is sent to the assigned email address of the username that was given as an input. The email contains a new password for the user.

Validate that user exists

In the below code, the username input is checked against the database.

An error message will be generated if the username given does not exist.

```
<?php

$errormsg= "DEF";

if(isset($_POST['submit'])){

//check if form was submitted
$username = $_POST['username'];

mysqli_select_db($conn,$database);
$sqlquery = "select * from customers where username='$username' ";
$check_user = mysqli_query($conn , $sqlquery) or
die("Database error detected: " . mysql_error());

$number = mysqli_num_rows( $check_user );
if ( $number == 0)  {
$errormsg = "NO"; }
```

If the user exists, fetch his data from database:

```
else {
```

```
//fetch user
$sqlquery = "select * from customers where username='$user
name' ";
$user = mysqli_query($conn , $sqlquery) or
die("Database error detected: " . mysql_error());

$fetches_user=mysqli_fetch_array($user);
```

Generate a new password with random numbers:

- Create 2 randoms numbers
- Shuffle them with the username
- Encrypt the newly created password
- Encrypt the password
- Store the password to the database

```
//create new password
$rand1=strval(rand(1000000,9999999));
$rand2=strval(rand(1000000,9999999));

$new_pass= $rand1;
$new_pass.= $fetches_user["username"];
$new_pass.= $rand2;
```

```
//encrypt new pass
```

```
$enc_password = hash('sha256', $new_pass);
```

```
//push new password to DB
```

```
$temp_user=$fetches_user["username"];
```

```
$push_pass=mysqli_query($conn,"UPDATE customers SET password=' $enc_password' where username=' $temp_user' ");
```

Create the email message and send it to the user:

```
//send email
```

```
$email=$fetches_user["email"];
```

```
$msg="This is MS FiT Care Gym. \n \n Your new password is: \n";
```

```
$msg.= $new_pass;
```

```
$msg.=" \n \n Thank you!";
```

```
$msg=wordwrap($msg,70);
```

```
mail($email,"Reset Password",$msg);
```

```
$errmsg = "YES";
```

```
}}
```

```
?>
```

BOOKING

The files and dependancies of the booking reside in folder /booking. All the actions implemented in index.php file, which includes the jquerys for datepicker.

Tech

Index.php uses a number of open source projects to work properly:

- [jQuery] - duh

features

Index.php implements a number of features such as:

for customer:

- select service
- select date
- select time
- cancel booking

for Physiotherapist:

- show all bookings of Physiotherapy

for Massage Therapist:

- show all bookings of Massage

for Admin:

- show all bookings per service
- clear all cancelled bookings
- clear all past bookings

Code and description

Index.php includes a jquery named datepicker for selecting date and time to book with a calendar interface.

```
//national holidays with fixed days
var holidays= [[1,1],[6,1], [25,3], [1,4], [1,5], [15,8],
[1,10], [28,10], [25,12], [26,12]];

//function to disable specific Dates
function DisableSpecificDates(date) {

for (var j = 0; j < holidays.length; j++) {
    var m= holidays[j][1]-1;
    var d=holidays[j][0];

    var dd=date.getDate();
    var mm = date.getMonth();

    //disable sundays
```

```

var day = date.getDay();
if ((day == 0)){
    return [false];}
//disable holidays
else if((d == dd)&&(m==mm)){

    return [false];
}
} return [true];
}

```

```

$(function() {
var dateToday = new Date();
$( "#from" ).datepicker({

    defaultDate: "+1w", //select default Date one week for
m today

    changeMonth: true, //allow all months

    numberOfMonths: 1, //show only one month in the cale
dar

    minDate: dateToday, //book only for dates grater t
han today

    onClose: function( selectedDate ) {
        },beforeShowDay: DisableSpecificDates
    });

```

```
});
```

Form to select “Service”, “Date” and “Time”

```
<form action="index.php" method="post">
    <label>Choose Service</label>
    //query to dynamically fill the drop bar with the
    services in the database
    <?php
        $sqls = "SELECT * FROM $servicestb ";
        $ress = mysqli_query($conn, $sqls);
        ?>
    <select class="form-control" id="choose-service" name="s
    ervice">
        <?php
            while ( $rows = mysqli_fetch_assoc($ress) ) {
                ?>
                <option type="radio" value="<?php echo $rows['tit
                le']?>" ><?php echo $rows['title'];
                }
                ?></option>
            </select>

    //input for date picked by the calendar
    <label>Choose Date</label>
    <input type='text' id="from" class="form-c
```

```
ontrol" name="day" />
        <input type="submit" class="btn btn-lg btn-pr
        imary btn-block round " name="available" type="submit" valu
        e="Check Availability"/>
    </form>
```

Check for availability

```
<?php
    // Fetch "Service" and "Time" selected
    $day = intval(strtotime(htmlspecialchars($_POST["day"]
));
    $service = htmlspecialchars($_POST["service"]);

    if(isset($_POST["available"])) {
        //fetch details of this service from database
        $sql3 = "SELECT * FROM $servicestb where title='$ser
        vice' ";
        $count2 = mysqli_query($conn, $sql3);
        $row3=mysqli_fetch_array($count2);
        ?>

        //form to select "Time"
        //for every time slot is a query to check if the spe
        cific time of the specific date is available according to
        the capacity of the service
```

```

<form action="index.php" method="post">
    <?php    $_POST["day"]=$day;
    $_POST["service"]=$service;
    $_SESSION["day"]=$day;
    $_SESSION["service"]=$service;

    ?>

    <label> Reservation Time</label>
    <select class="form-control" id="choose-service" name="time">
        <?php
            $sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0  and time='09:00-10:00' and day=
'$day' ";
            $count = mysqli_query($conn, $sql2);
            $row2 = mysqli_fetch_array($count);

            if (!( $row2[0]>=$row3['capacity'] )){ ?>

                <option selected="selected">09:00-10:00</option>
        <?php } ?>

        <?php
            $sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0  and time='10:00-11:00' and day=
'$day' ";
            $count = mysqli_query($conn, $sql2);
            $row2 = mysqli_fetch_array($count);

```

```
if (!($row2[0]>=$row3['capacity'])) { ?>
```

```
<option>10:00-11:00</option> <?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='11:00-12:00' and day=
'$day' ";
```

```
$count = mysqli_query($conn, $sql2);
```

```
$row2 = mysqli_fetch_array($count);
```

```
if (!($row2[0]>=$row3['capacity'])) { ?>
```

```
<option>11:00-12:00</option> <?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='12:00-13:00' and day=
'$day' ";
```

```
$count = mysqli_query($conn, $sql2);
```

```
$row2 = mysqli_fetch_array($count);
```

```
if (!($row2[0]>=$row3['capacity'])) { ?>
```

```
<option>12:00-13:00</option> <?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='13:00-14:00' and day=
'$day' ";
```

```
$count = mysqli_query($conn, $sql2);
```

```
$row2 = mysqli_fetch_array($count);
```

```
if (!( $row2[0]>=$row3['capacity'] )){ ?>
```

```
<option>13:00-14:00</option><?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='14:00-15:00' and day=
'$day' ";
```

```
$count = mysqli_query($conn, $sql2);
```

```
$row2 = mysqli_fetch_array($count);
```

```
if (!( $row2[0]>=$row3['capacity'] )){ ?>
```

```
<option>14:00-15:00</option> <?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='15:00-16:00' and day=
'$day' ";
```

```
$count = mysqli_query($conn, $sql2);
```

```
$row2 = mysqli_fetch_array($count);
```



```
if (!( $row2[0]>=$row3['capacity'] )){ ?>
```

```
<option>15:00-16:00</option> <?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='16:00-17:00' and day=
'$day' ";
```

```
$count = mysqli_query($conn, $sql2);
```

```
$row2 = mysqli_fetch_array($count);
```

```
if (!( $row2[0]>=$row3['capacity'] )){ ?>
```

```
<option>16:00-17:00</option> <?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='17:00-18:00' and day=
'$day' ";
```

```
$count = mysqli_query($conn, $sql2);
```

```
$row2 = mysqli_fetch_array($count);
```

```
if (!( $row2[0]>=$row3['capacity'] )){ ?>
```

```
<option>17:00-18:00</option> <?php } ?>
```

```
<?php
```

```
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
```

```

'$service' and canceled=0 and time='18:00-19:00' and day=
'$day' ";

$count = mysqli_query($conn, $sql2);
$row2 = mysqli_fetch_array($count);

if (!($row2[0]>=$row3['capacity'])) { ?>

<option>18:00-19:00</option> <?php } ?>

<?php
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='19:00-20:00' and day=
'$day' ";

$count = mysqli_query($conn, $sql2);
$row2 = mysqli_fetch_array($count);

if (!($row2[0]>=$row3['capacity'])) { ?>

<option>19:00-20:00</option> <?php } ?>

<?php
$sql2 = "SELECT COUNT(*) from $booktb WHERE service=
'$service' and canceled=0 and time='20:00-21:00' and day=
'$day' ";

$count = mysqli_query($conn, $sql2);
$row2 = mysqli_fetch_array($count);

if (!($row2[0]>=$row3['capacity'])) { ?>

```

```
<option>20:00-21:00</option> <?php } ?>
```

```
</select>
```

```
<input name="book" type="submit" value="Book" class="btn btn-lg btn-success btn-block round" />
</form>
```

Cancel a booking

form to get the Id of the booking

```
<form action="index.php" method="post">
  <input type='text' id="from" class="form-control" name="id" />
  <input name="cancel" type="submit" class="btn btn-lg btn-danger btn-block round" value="Cancel" />
</form>
```

Permissions

```
//Fetch "role" and "username"
$role=$_SESSION['role'];
$user=$_SESSION['username'];
```

```
if (($role=="Admin")||($role=="physiotherapist")||($role=="MessageTherapist")){

//if user is a Physiotherapist
if ($role=="physiotherapist") {
//query to show total amount of booking for physiotherapy
    $sql9 = "SELECT COUNT(*) from $booktb WHERE service='Physiotherapy'and canceled=0 ";
    $use = mysqli_query($conn, $sql9);
    $row9 = mysqli_fetch_array($use);
    $userows=$row9[0];
    //query to show all valid bookings of physiotherapy
    $sql10 = "SELECT * FROM $booktb WHERE service='Physiotherapy' and canceled=0 Order by day";

    $result10 = mysqli_query($conn, $sql10);

    $c=0;
    //if there are booking for physiotherapy, show them
    if ($userows > 0) {

        echo "Bookings of Physiotherapy: ";

        echo "\tBookID\tUsername\tService\tDate\tTime\t";

        while ($row10 = mysqli_fetch_assoc($result10)) {

            echo $row10["id"]."\t".$row10["service"]."\t";
```

```

        echo date('m/d/Y', $row10["day"]);
        echo "\t". $row10["time"];

        $c++;
    }

    echo "Total ".$c." bookings";
} else {
    echo "<h3>No Bookings of Physiotherapy</h3>";
}

// if user is a Massage Therapist
}else if ($role=="MassageTherapist") {
//query to show total amount of booking for massage
    $sql9 = "SELECT COUNT(*) from $booktb WHERE service='Ma
ssage' and canceled=0";

    $use = mysqli_query($conn, $sql9);
    $row9 = mysqli_fetch_array($use);
    $userows=$row9[0];

    //query to show all valid bookings of massage
    $sql10 = "SELECT * FROM $booktb WHERE service='Massage'
and canceled=0 Order by day";

    $result10 = mysqli_query($conn, $sql10);

    $c=0;

    //if there are booking for massage, show them
    if ($userows > 0) {

```

```

        echo "Bookings of Massage: ";

        echo "\tBookID\tUsername\tService\tDate\tTime\t";

        while ($row10 = mysqli_fetch_assoc($result10)) {

            echo $row10["id"]."\t".$row10["service"]."\t";
            echo date('m/d/Y',$row10["day"]);
            echo "\t".$row10["time"];

            $c++;
        }

        echo "Total ".$c." bookings";
    } else {
        echo "<h3>No Bookings of Massage</h3>";
    }
}

```

Permissions for Admin

form to get input for Clear Past Bookings

```

<form method="post" class="py-3" action="index.php">
    <input name="past" type="submit" class="btn btn-

```

```
lg btn-danger round " value="Clear Past Bookings" />
</form>
```

form to get input for Clear Cancelled Bookings

```
<form method="post" class="py-2" action="index.php">
    <input name="delete" type="submit" class="btn btn-
n-lg btn-success round " value="Clear Cancelled Bookings!"
/>
</form>
```

Admin

```
<?php if ($role=="Admin"){
    //query to select all services
    $sqls = "SELECT * FROM $servicestb ";
    $serv = mysqli_query($conn, $sqls);
    //for all services
    while ( $rowser = mysqli_fetch_assoc($serv) ) {
        $servicet= $rowser['title'];
        //query to count the booking of this service
        $sql9 = "SELECT COUNT(*) from $booktb WHERE servic
e='$servicet' ";
        $use = mysqli_query($conn, $sql9);
        $row9 = mysqli_fetch_array($use);
```

```

$userrows=$row9[0];

//query to select all data ordered by day for this
service

$sql10 = "SELECT * FROM $booktb WHERE service='$se
rvicet' Order by day";

$result10 = mysqli_query($conn, $sql10);

$c=0;

//if there are bookings for this service, show the
m

if ($userrows > 0) {
    echo $servicet;
    while ($row10 = mysqli_fetch_assoc($result10))
    {
        $row10["id"]."\t".$row10["username"]."\t"
.$row10["service"]."\t";
        echo date('m/d/Y',$row10["day"]);
        echo "\t".$row10["time"]."\t".$row10["canc
eled"];

        $c++;
    }
    echo "Total ".$c." bookings";
} }

```

Action: Booking

if the book button is pushed, the following actions and checks are made

```

<?php
    if(isset($_POST["book"])) {
        //Fetch "current time","username","service","day"and
        "time"

        $time = htmlspecialchars($_POST["time"]);
        $username = $_SESSION['username'];
        $service=$_SESSION["service"];
        $day=$_SESSION["day"];
        $_SESSION["time"]=$time;

        // prevent double booking
        //query to check if the user has an other booking
        for this specific date and time

        $sql4 = "SELECT COUNT(*) from $booktb WHERE servi
ce='$service' and canceled=0 and time='$time' and day='$d
ay' and username='$username' ";
        $count4 = mysqli_query($conn, $sql4);
        $row4 = mysqli_fetch_array($count4);
        if($row4[0]>0){
            $_SESSION['error']="Unfortunately you booked
an other service for this time and date.";
        }

        //query to insert the new booking to the database
        $sql = "INSERT INTO $booktb (username, service, da

```

```

y, time, canceled)

        VALUES ('$username', '$service', '$day', '$time', 0)";

//send confermation email to user
$corday= date('m/d/Y',$day);
$email=$_SESSION['email'];
$sql10 = "SELECT * FROM $booktb WHERE username='$username' and
        day='$day' and time='$time' and service='$service'";

$re = mysqli_query($conn, $sql10);
$id=mysqli_fetch_array($re);

$msg= "This is MS FiT Care Gym. \n\nYour booking for ".$service." on ".$corday." at ".$time." with ID:".$id["id"]." was succesfully registered!";
$msg=wordwrap($msg,70);
mail($email,"Booking Confirmation ", $msg);

} else {
        echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

//print appropriate message and refresh website
$_SESSION['error']="Booking succeed.";
        echo '<meta http-equiv="refresh" content="0">';

```

```
}
```

```
?>
```

Action: Cancel Booking

if the cancel button is pushed, the following actions and checks are made

```
<?php
if(isset($_POST["cancel"])) {
//Fetch "id","username" and "role"
    $id = intval(htmlspecialchars($_POST["id"]));
    $username=$_SESSION['username'];
    $role=$_SESSION['role'];

    //check if user cancels HIS booking

    $sql0 = "SELECT COUNT(*) from $booktb WHERE id = $
id and username='$username'";
    $count0 = mysqli_query($conn, $sql0);
    $row0 = mysqli_fetch_array($count0);

    $sql5= "SELECT service from $booktb where id=$id";
    $serv = mysqli_query($conn, $sql5);
    $row5 = mysqli_fetch_array($serv);

    //check if user is Admin,Physiotherapist or Massage
```

Therapist

```
if (($row0[0]==1)||($role=="Admin"))||(($row5[0]=="
Physiotherapy")&&($role=="physiotherapist"))||(($row5[0]=="
Massage")&&($role=="MassageTherapist"))){
    //query to update the booking to cancelled
    $sql = "UPDATE $booktb SET canceled=1 WHERE id = $
id";
    //print appropriate message and refresh website
    if (mysqli_query($conn, $sql)) {
        $_SESSION['error']="Booking cancelled.";
    }
    else {
        echo "Error: " . $sql . "<br>" . mysqli_error(
$conn);
    }
}
else{
    $_SESSION['error']="This is not your booking"
;
}
echo '<meta http-equiv="refresh" content="0">';
}
?>
```

Action: Clear Cancelled Bookings

if the Clear Cancelled Bookings button is pushed, the following actions

and checks are made

```
<?php if(isset($_POST["delete"])) {  
    //query to count the amount of cancelled bookings  
    $sql6 = "SELECT COUNT(*) from $booktb WHERE cancel  
ed=1 ";  
    $canc = mysqli_query($conn, $sql6);  
    $row6 = mysqli_fetch_array($canc);  
    $cancelled=$row6[0];  
    //query to select all cancelled bookings  
    $sql7 = "SELECT * FROM $booktb WHERE canceled=1";  
    $resu = mysqli_query($conn, $sql7);  
    $c=0;  
    if ($cancelled > 0) {  
        while($row = mysqli_fetch_assoc($resu)){  
            //query to delete this specific cancelled  
booking  
            $sql8 = "DELETE FROM $booktb WHERE canceled=1 ";  
            if (mysqli_query($conn, $sql8)) {  
                $c++;  
            }  
            else {  
                echo "Error: " . $sql8 . "<br>" . mysql  
li_error($conn);  
            }  
        }  
    }  
}
```

```

        //print appropriate message and refresh website
        $_SESSION['error'] = "Cancelled Bookings are Cleared, total bookings cancelled: $c";
    } else {
        $_SESSION['error'] = "No bookings deleted";
    }
    echo '<meta http-equiv="refresh" content="0">';

} ?>

```

Action: Clear Past Bookings

if the Clear Past Bookings button is pushed, the following actions and checks are made

```

<?php    if(isset($_POST["past"])) {
        $timenow = time(); //fetch current time
        ////query to count the amount of past bookings
        $sql6 = "SELECT COUNT(*) from $booktb WHERE day<'$timenow'";
        $canc = mysqli_query($conn, $sql6);
        $row6 = mysqli_fetch_array($canc);
        $cancelled=$row6[0];
        //query to select all past bookings
        $sql7 = "SELECT * FROM $booktb WHERE day<'$timenow'";
        $resu = mysqli_query($conn, $sql7);
        $c=0;
    }
}

```

```
if ($cancelled > 0) {  
    while($row = mysqli_fetch_assoc($resu)){  
        $sql8 = "DELETE FROM $booktb WHERE day<'$timen  
ow'";  
        if (mysqli_query($conn, $sql8)) {  
            $c++;  
        }  
        else {  
            echo "Error: " . $sql8 . "<br>" . mysqli_e  
rror($conn);  
        }  
    }  
    //print appropriate message and refresh website  
    $_SESSION['error'] ="Past Bookings are Cleared, to  
tal bookings cancelled: $c";  
} else {  
    $_SESSION['error']="No past Bookings";  
}  
echo '<meta http-equiv="refresh" content="0">';  
} ?>
```