LAU

الجامعة اللبنانيّة الأميركيّة

Lebanese American University

Phase 3: CSC490

Football Club Management System

OUTSIDE THE PITCH

By:

Hussein Ali Ahmed (ID: 202104969)
Bilal Delbani (ID: 202104998)
Mohammad Al Fallah (ID: 202105098)
Omar Fakhredin (ID: 202101582)

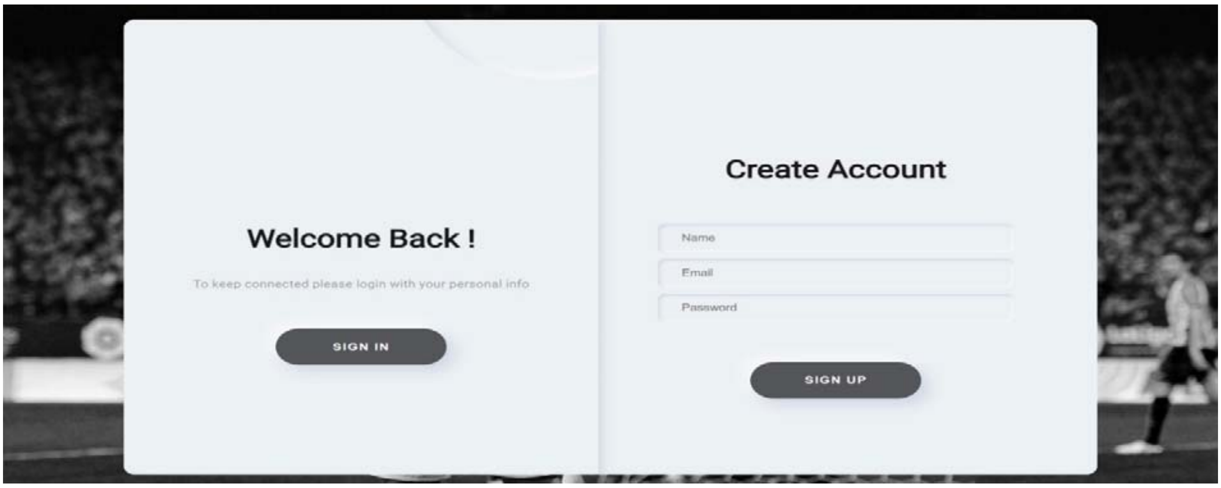
Introduction:

Outside the pitch is a piece of software developed to help football teams run their day-to-day operations more successfully. It provides a variety of functionalities to club administrators, players, and supporters, making it easier to communicate, organize, and execute activities. Club administrators can supervise the club's finances, arrange matches, handle player contracts, and more. Coaches can utilize the software to evaluate game films, design training plans, and track player performance. Players can use the software to monitor their timetables, communicate with coaches and teammates, and track their personal development. Finally, supporters can utilize the software to purchase merchandise, products, and receive match schedule and result updates, and engage with other fans. This software was implemented as a website using HTML, CSS, JavaScript, and PHP to link to a database managed by MySQL. The functionalities that will be shown in this document are the following:

- Sign up
- Log in
- Purchase products and merchandise
- Manager updating player information

Functionalities:

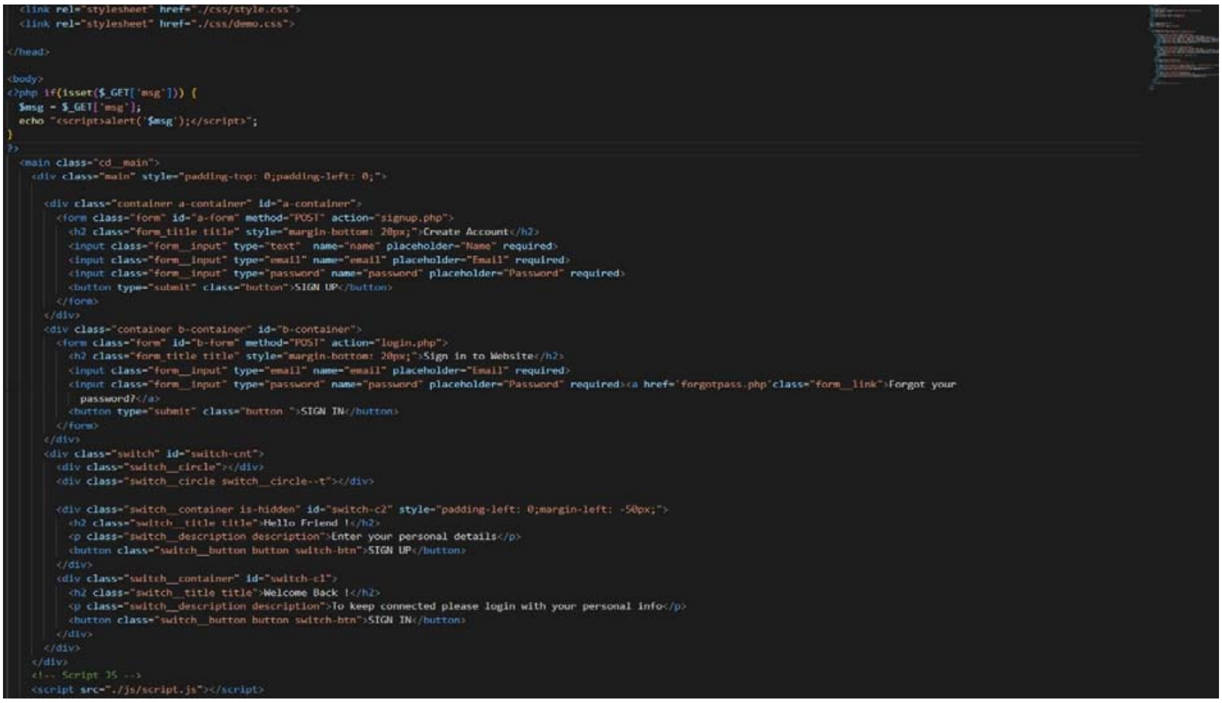
Sign up:



Welcome Back !

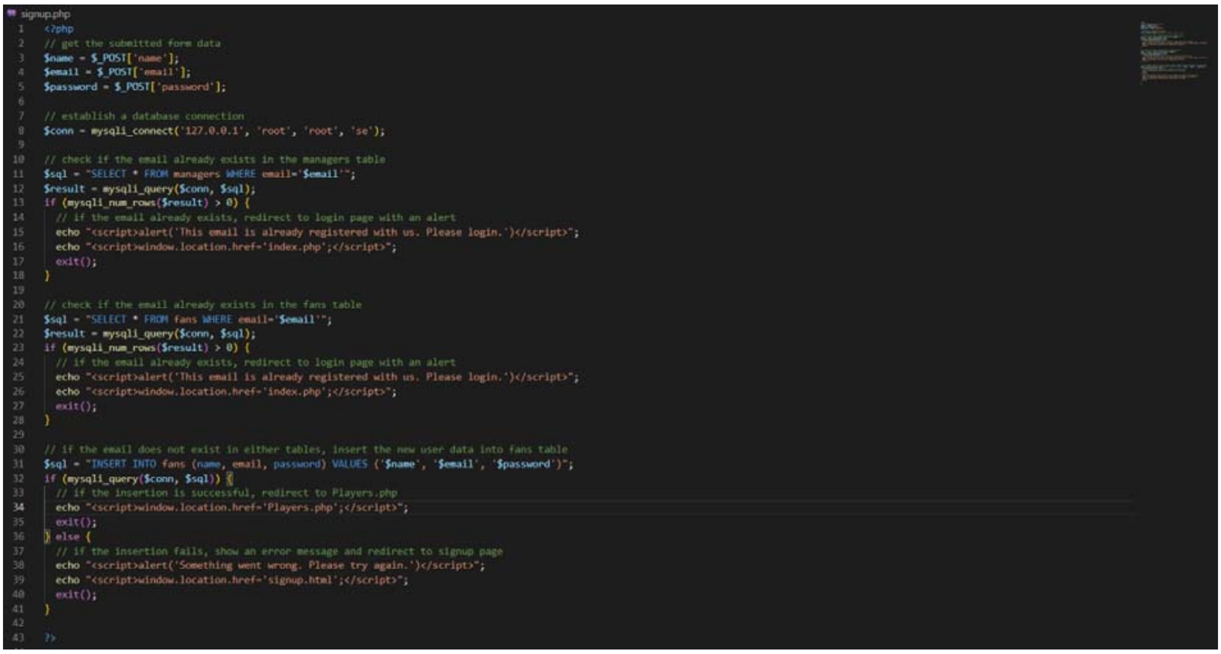
To keep connected please login with your personal info

Create Account



```
  <link rel="stylesheet" href="./css/style.css">
  <link rel="stylesheet" href="./css/demo.css">

</head>

<body>
<?php if(isset($_GET['msg'])) {
  $msg = $_GET['msg'];
  echo "<script>alert('$msg');</script>";
}
?>
  <main class="cd__main">
    <div class="main" style="padding-top: 0;padding-left: 0;">

      <div class="container a-container" id="a-container">
        <form class="form" id="a-form" method="POST" action="signup.php">
          <h2 class="form_title title" style="margin-bottom: 20px;">Create Account</h2>
          <input class="form__input" type="text"  name="name" placeholder="Name" required>
          <input class="form__input" type="email" name="email" placeholder="Email" required>
          <input class="form__input" type="password" name="password" placeholder="Password" required>
          <button type="submit" class="button">SIGN UP</button>
        </form>
      </div>
      <div class="container b-container" id="b-container">
        <form class="form" id="b-form" method="POST" action="login.php">
          <h2 class="form_title title" style="margin-bottom: 20px;">Sign in to Website</h2>
          <input class="form__input" type="email" name="email" placeholder="Email" required>
          <input class="form__input" type="password" name="password" placeholder="Password" required><a href='forgotpass.php'class="form__link">Forgot your
           password?</a>
          <button type="submit" class="button ">SIGN IN</button>
        </form>
      </div>
      <div class="switch" id="switch-cnt">
        <div class="switch__circle"></div>
        <div class="switch__circle switch__circle--t"></div>

        <div class="switch__container is-hidden" id="switch-c2" style="padding-left: 0;margin-left: -50px;">
          <h2 class="switch__title title">Hello Friend !</h2>
          <p class="switch__description description">Enter your personal details</p>
          <button class="switch__button button switch-btn">SIGN UP</button>
        </div>
        <div class="switch__container" id="switch-c1">
          <h2 class="switch__title title">Welcome Back !</h2>
          <p class="switch__description description">To keep connected please login with your personal info</p>
          <button class="switch__button button switch-btn">SIGN IN</button>
        </div>
      </div>
    </div>
    <!-- Script JS -->
    <script src="./js/script.js"></script>
```

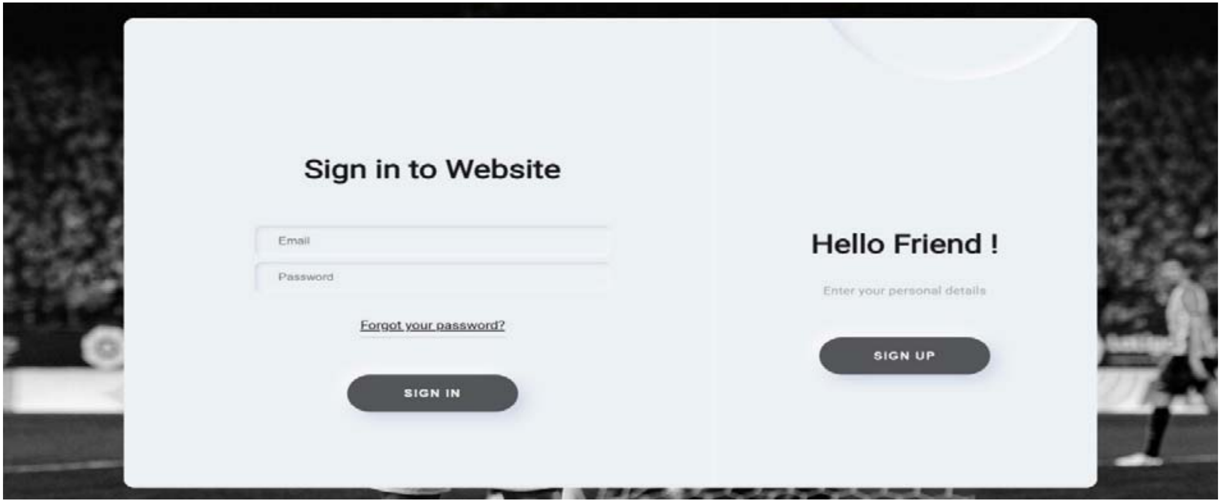
This provides a form for signing up and a form for logging in. When toggled, the form and the fields the user wants will show. The form for the signup page has a name, email, and password fields. The action taken when signing up will be handled by the file signup.php.



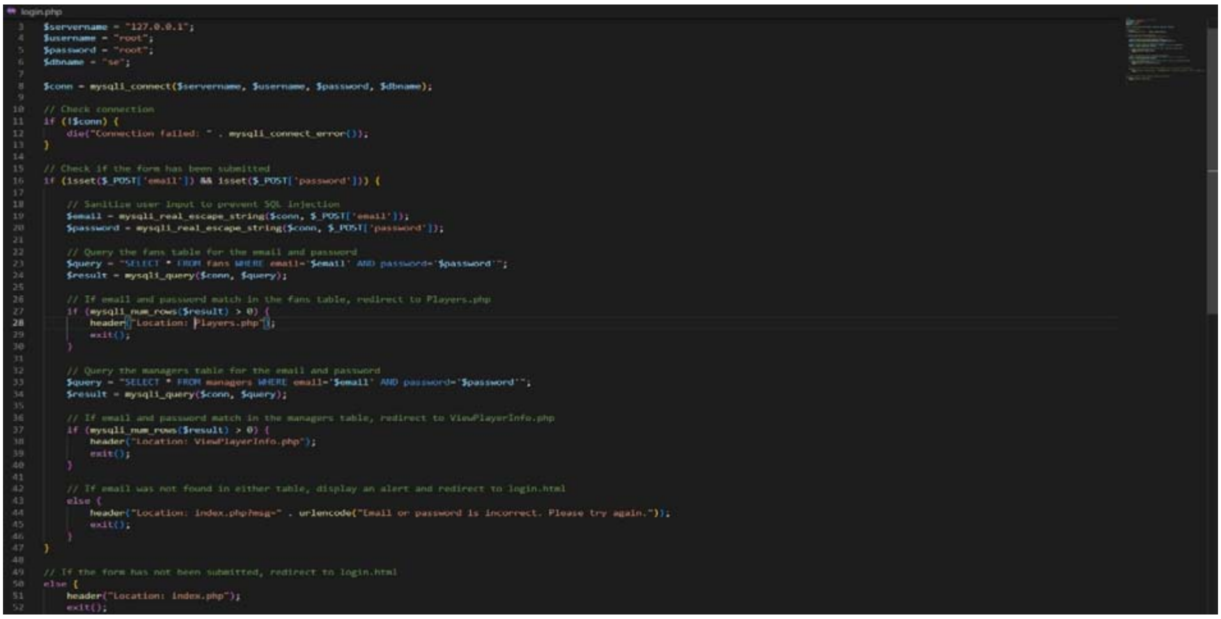
```
<?php
// get the submitted form data
$name = $_POST['name'];
$email = $_POST['email'];
$password = $_POST['password'];

// establish a database connection
$conn = mysqli_connect('127.0.0.1', 'root', 'root', 'se');

// check if the email already exists in the managers table
$sql = "SELECT * FROM managers WHERE email='$email'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
  // if the email already exists, redirect to login page with an alert
  echo "<script>alert('This email is already registered with us. Please login.')</script>";
  echo "<script>window.location.href='index.php';</script>";
  exit();
}

// check if the email already exists in the fans table
$sql = "SELECT * FROM fans WHERE email='$email'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
  // if the email already exists, redirect to login page with an alert
  echo "<script>alert('This email is already registered with us. Please login.')</script>";
  echo "<script>window.location.href='index.php';</script>";
  exit();
}

// if the email does not exist in either tables, insert the new user data into fans table
$sql = "INSERT INTO fans (name, email, password) VALUES ('$name', '$email', '$password')";
if (mysqli_query($conn, $sql)) {
  // if the insertion is successful, redirect to Players.php
  echo "<script>window.location.href='Players.php';</script>";
  exit();
} else {
  // if the insertion fails, show an error message and redirect to signup page
  echo "<script>alert('Something went wrong. Please try again.')</script>";
  echo "<script>window.location.href='signup.html';</script>";
  exit();
}

?>
```

This code takes the email inputted by the user. If mysqli_num_rows return greater than 0 for the manager or fans table this means that this email is already used by another person and an alert will be shown for the person trying to use it. Else, the name, email, and password will be inserted into the fan's table. If the insertion was successful, the user will be directed to the Players page. Otherwise, an alert will be shown and the user will be redirected to the signup page.

Log in:

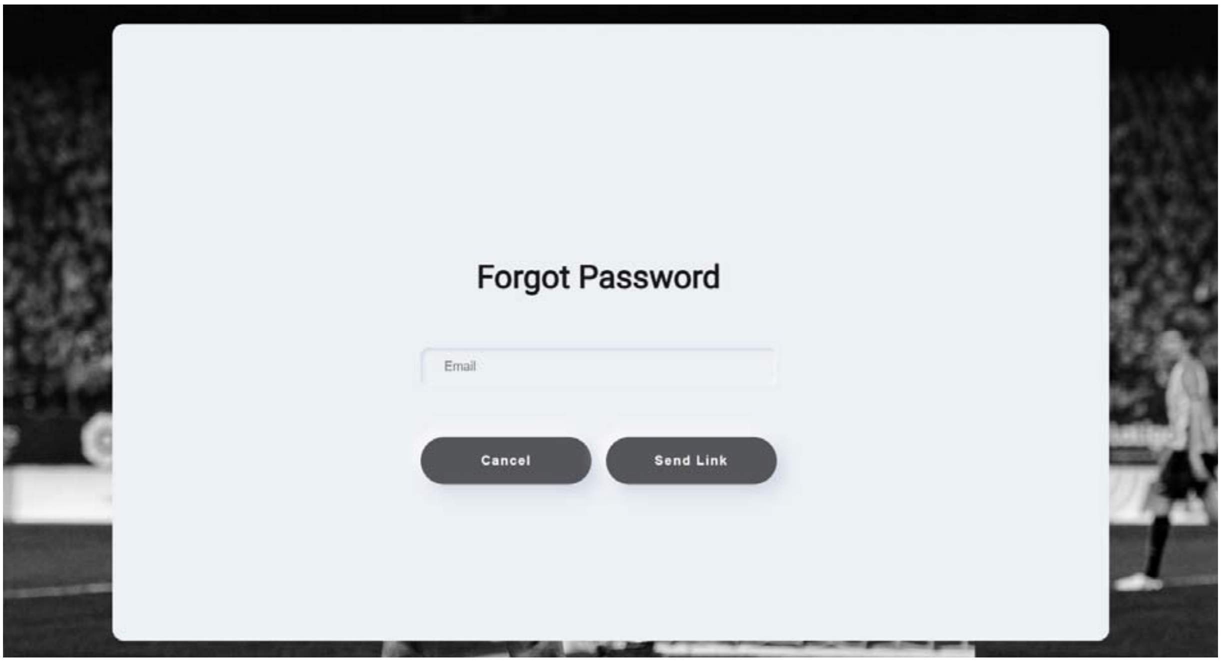


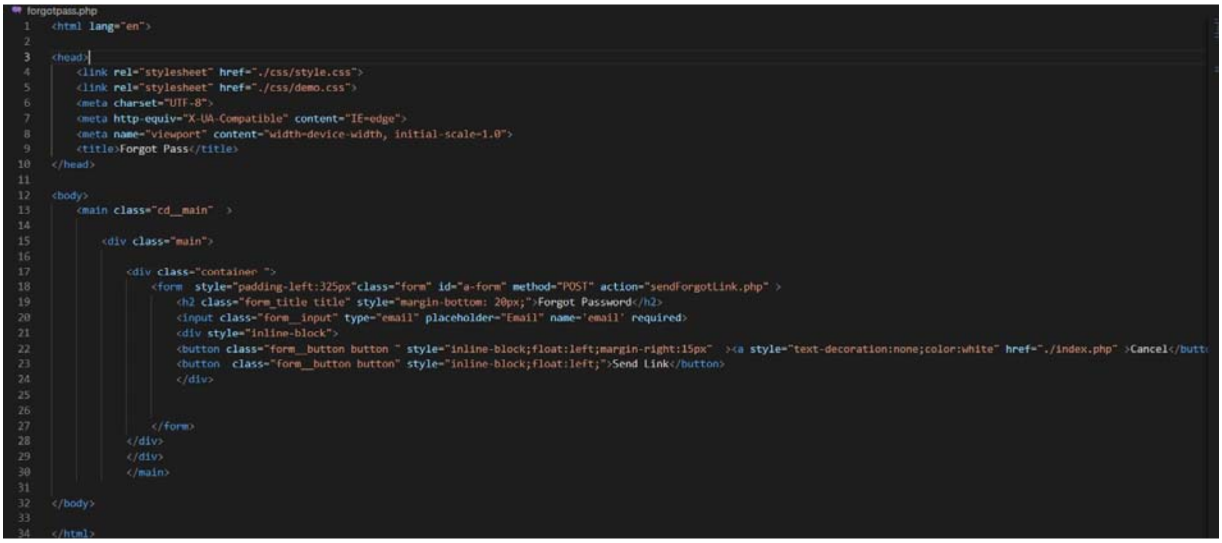
This form is shown when toggled on the first login page as shown in the first HTML, CSS, and JS code. The file responsible for taking action for this form is the login.php.



```php
$servername = "127.0.0.1";
$username = "root";
$password = "root";
$dbname = "se";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Check if the form has been submitted
if (isset($_POST['email']) && isset($_POST['password'])) {

    // Sanitize user input to prevent SQL injection
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $password = mysqli_real_escape_string($conn, $_POST['password']);

    // Query the fans table for the email and password
    $query = "SELECT * FROM fans WHERE email='$email' AND password='$password'";
    $result = mysqli_query($conn, $query);

    // If email and password match in the fans table, redirect to Players.php
    if (mysqli_num_rows($result) > 0) {
        header("Location: Players.php");
        exit();
    }

    // Query the managers table for the email and password
    $query = "SELECT * FROM managers WHERE email='$email' AND password='$password'";
    $result = mysqli_query($conn, $query);

    // If email and password match in the managers table, redirect to ViewPlayerInfo.php
    if (mysqli_num_rows($result) > 0) {
        header("Location: ViewPlayerInfo.php");
        exit();
    }

    // If email was not found in either table, display an alert and redirect to login.html
    else {
        header("Location: index.php?msg=" . urlencode("Email or password is incorrect. Please try again."));
        exit();
    }
}

// If the form has not been submitted, redirect to login.html
else {
    header("Location: index.php");
    exit();
```

When the user presses sign in the email and password will be queried to the database to check whether they are present in the fans or manager table. The mysqli_num_rows returns the number of rows in the database. If the number is greater than 0 in the fans table, the user will be directed to the Players.php page. If the number is positive in the manager's table, the user will be directed to the ViewPlayersInfo.php page. Otherwise, an alert will be shown and the user will return to the login page. There is also the forgot password page:

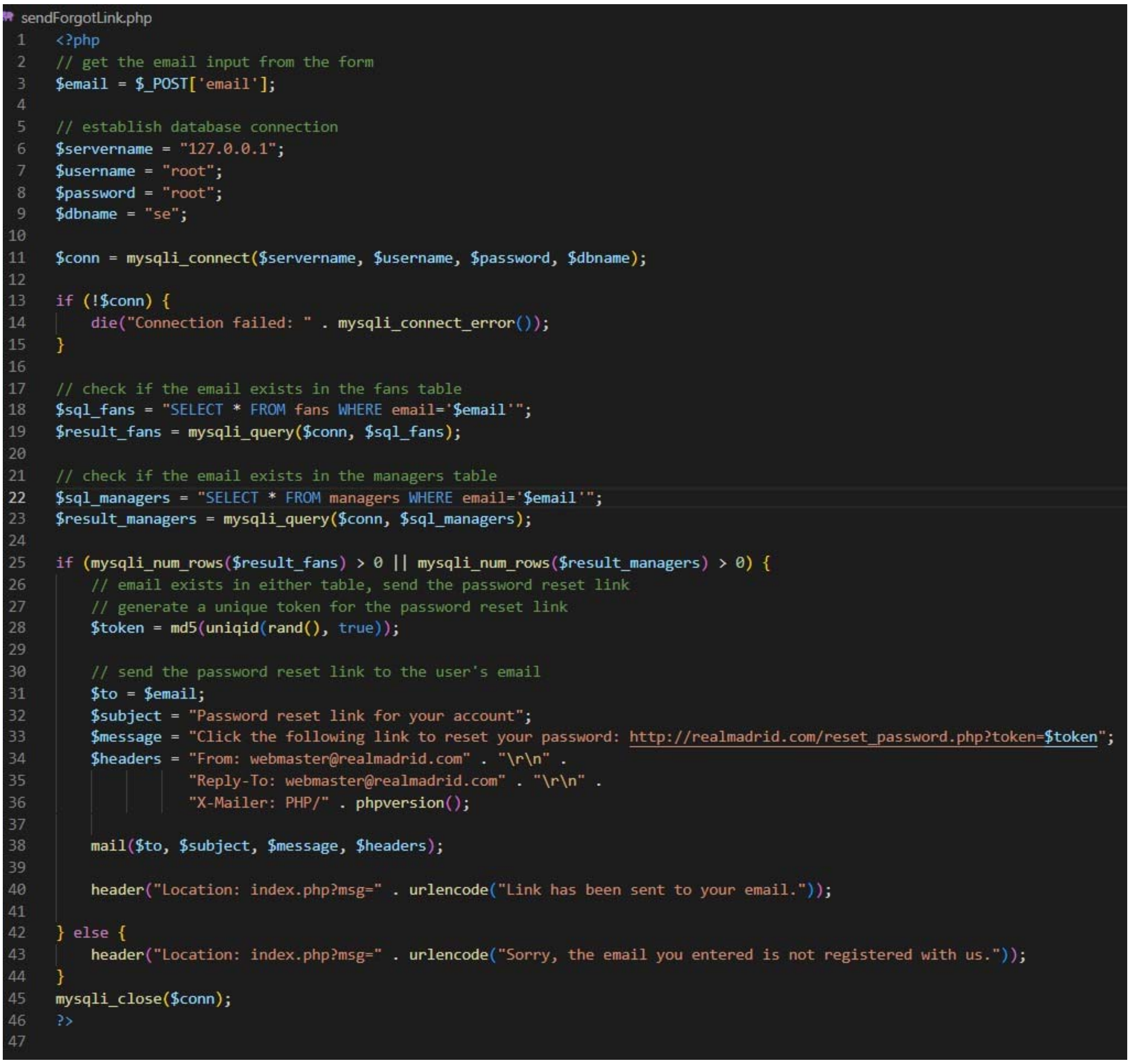



forgotpass.php

```
<html lang="en">

<head>
    <link rel="stylesheet" href="./css/style.css">
    <link rel="stylesheet" href="./css/demo.css">
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Forgot Pass</title>
</head>

<body>
    <main class="cd__main"  >

        <div class="main">

            <div class="container ">
                <form  style="padding-left:325px"class="form" id="a-form" method="POST" action="sendForgotLink.php" >
                    <h2 class="form_title title" style="margin-bottom: 20px;">Forgot Password</h2>
                    <input class="form__input" type="email" placeholder="Email" name='email' required>
                    <div style="inline-block">
                    <button class="form__button button " style="inline-block;float:left;margin-right:15px"  ><a style="text-decoration:none;color:white" href="./index.php" >Cancel</butt
                    <button  class="form__button button" style="inline-block;float:left;">Send Link</button>
                    </div>


                </form>
            </div>
            </div>
            </main>

</body>

</html>
```

It provides a form with an email field to retrieve the password for the email.

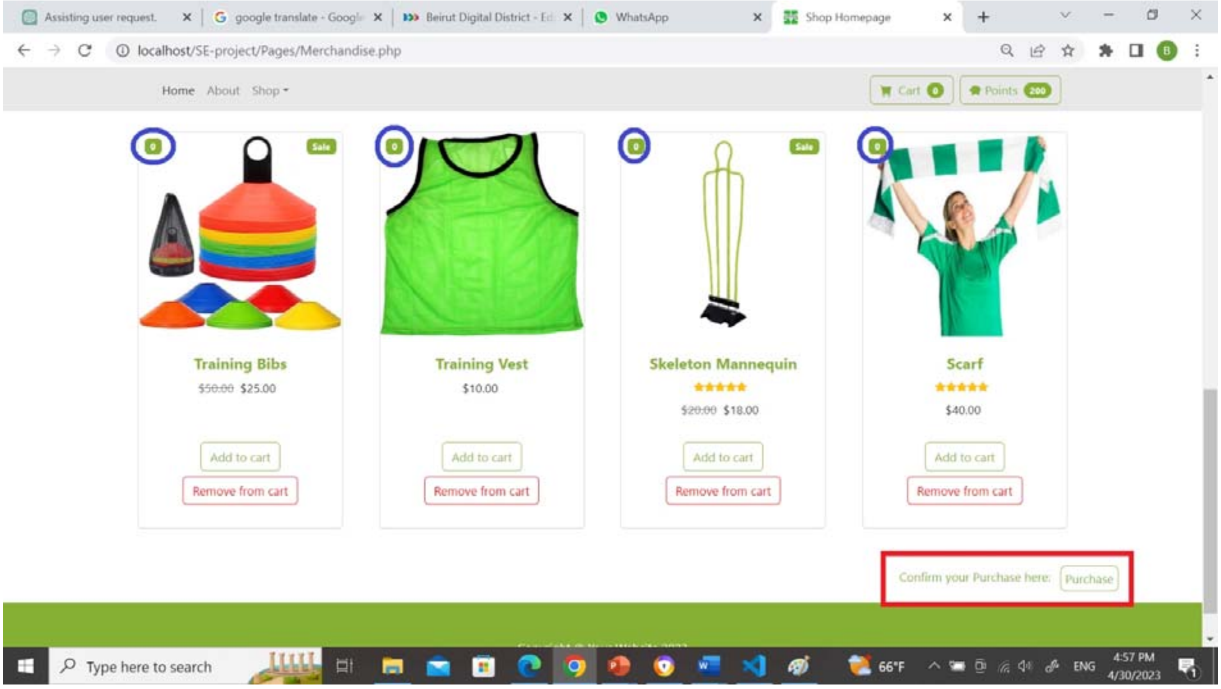
The PHP file responsible for taking action for this page is the sendForgotLink.php.

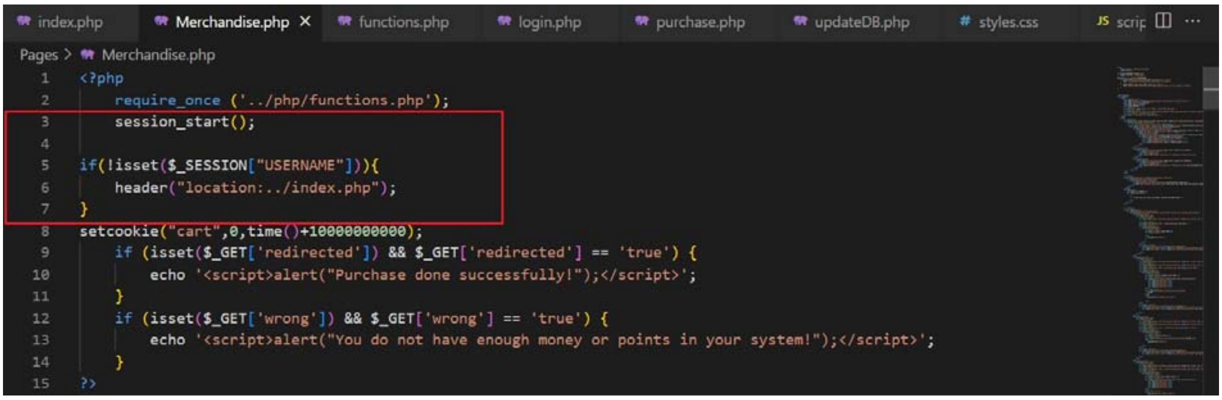

sendForgotLink.php

```php
<?php
// get the email input from the form
$email = $_POST['email'];

// establish database connection
$servername = "127.0.0.1";
$username = "root";
$password = "root";
$dbname = "se";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// check if the email exists in the fans table
$sql_fans = "SELECT * FROM fans WHERE email='$email'";
$result_fans = mysqli_query($conn, $sql_fans);

// check if the email exists in the managers table
$sql_managers = "SELECT * FROM managers WHERE email='$email'";
$result_managers = mysqli_query($conn, $sql_managers);

if (mysqli_num_rows($result_fans) > 0 || mysqli_num_rows($result_managers) > 0) {
    // email exists in either table, send the password reset link
    // generate a unique token for the password reset link
    $token = md5(uniqid(rand(), true));

    // send the password reset link to the user's email
    $to = $email;
    $subject = "Password reset link for your account";
    $message = "Click the following link to reset your password: http://realmadrid.com/reset_password.php?token=$token";
    $headers = "From: webmaster@realmadrid.com" . "\r\n" .
               "Reply-To: webmaster@realmadrid.com" . "\r\n" .
               "X-Mailer: PHP/" . phpversion();

    mail($to, $subject, $message, $headers);

    header("Location: index.php?msg=" . urlencode("Link has been sent to your email."));

} else {
    header("Location: index.php?msg=" . urlencode("Sorry, the email you entered is not registered with us."));
}
mysqli_close($conn);
?>
```

This code will search for the fan or the manager that has the email entered in the form. If $result_fans or $result_managers are greater than 0, the user’s email was found, and he will be sent a link to reset his password. Else an alert will be sent and the user will not be able to reset the password.

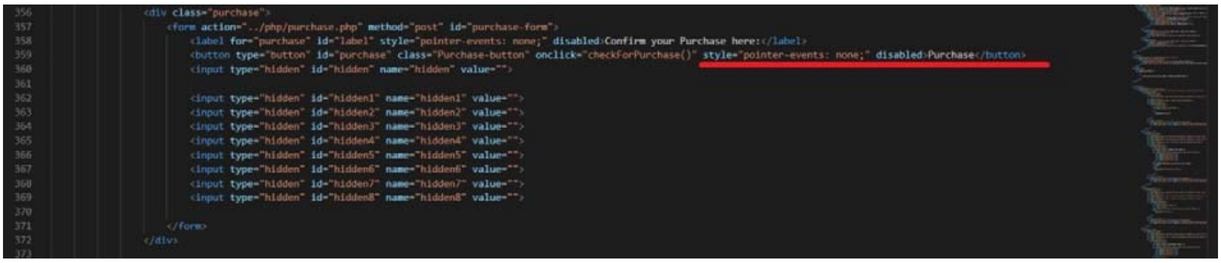
Purchasing products:





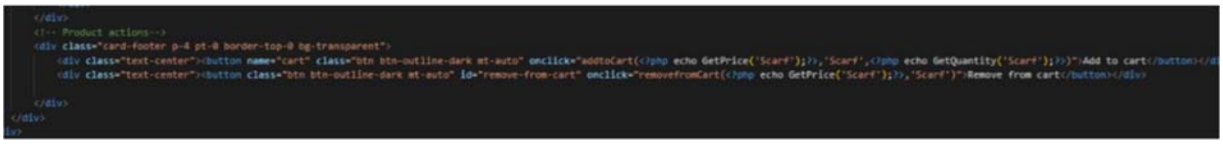
```php
<?php
    require_once ('../php/functions.php');
    session_start();

if(!isset($_SESSION["USERNAME"])){
    header("location:../index.php");
}
setcookie("cart",0,time()+10000000000);
    if (isset($_GET['redirected']) && $_GET['redirected'] == 'true') {
        echo '<script>alert("Purchase done successfully!");</script>';
    }
    if (isset($_GET['wrong']) && $_GET['wrong'] == 'true') {
        echo '<script>alert("You do not have enough money or points in your system!");</script>';
    }
?>
```

If the user entering has no valid username, he can’t purchase for security reasons.



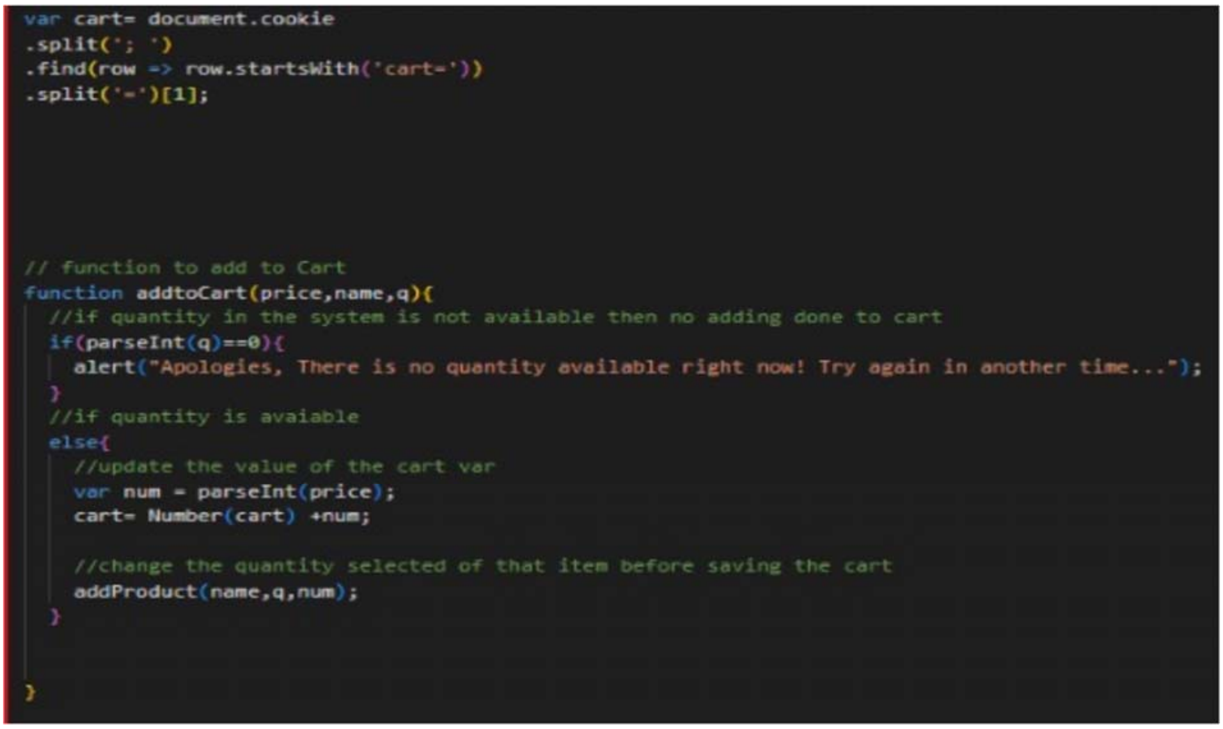
```html
<div class="purchase">
    <form action="../php/purchase.php" method="post" id="purchase-form">
        <label for="purchase" id="label" style="pointer-events: none;" disabled>Confirm your Purchase here:</label>
        <button type="button" id="purchase" class="Purchase-button" onclick="checkForPurchase()" style="pointer-events: none;" disabled>Purchase</button>
        <input type="hidden" id="hidden" name="hidden" value="">

        <input type="hidden" id="hidden1" name="hidden1" value="">
        <input type="hidden" id="hidden2" name="hidden2" value="">
        <input type="hidden" id="hidden3" name="hidden3" value="">
        <input type="hidden" id="hidden4" name="hidden4" value="">
        <input type="hidden" id="hidden5" name="hidden5" value="">
        <input type="hidden" id="hidden6" name="hidden6" value="">
        <input type="hidden" id="hidden7" name="hidden7" value="">
        <input type="hidden" id="hidden8" name="hidden8" value="">

    </form>
</div>
```

If the user has no items in the cart he can’t click on the purchase button. Otherwise, it will be green and clickable.



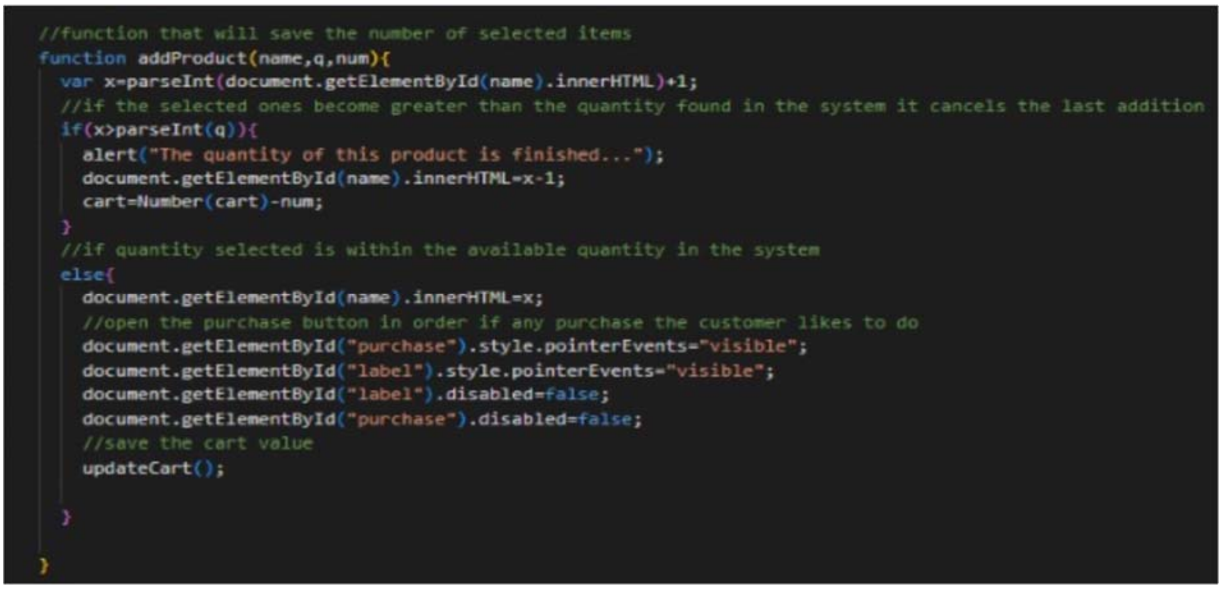
```
    </div>
    <!-- Product actions-->
    <div class="card-footer p-4 pt-0 border-top-0 bg-transparent">
        <div class="text-center"><button name="cart" class="btn btn-outline-dark mt-auto" onclick="addtoCart(<?php echo GetPrice('Scarf');?>,'Scarf',<?php echo GetQuantity('Scarf');?>)">Add to cart</button></d
        <div class="text-center"><button class="btn btn-outline-dark mt-auto" id="remove-from-cart" onclick="removefromCart(<?php echo GetPrice('Scarf');?>,'Scarf')">Remove from cart</button></div>

    </div>
  </div>
iv>
```

This is for the add-to-cart and remove-from-cart functionalities which are then handled using JavaScript and PHP codes.



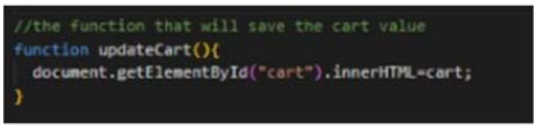
```
var cart= document.cookie
.split('; ')
.find(row => row.startsWith('cart='))
.split('=')[1];






// function to add to Cart
function addtoCart(price,name,q){
  //if quantity in the system is not available then no adding done to cart
  if(parseInt(q)==0){
    alert("Apologies, There is no quantity available right now! Try again in another time...");
  }
  //if quantity is avaiable
  else{
    //update the value of the cart var
    var num = parseInt(price);
    cart= Number(cart) +num;

    //change the quantity selected of that item before saving the cart
    addProduct(name,q,num);
  }


}
```

If the quantity of the product purchased is zero the user will be alerted. Otherwise, the money that should be paid will be updated as well as the quantity of the selected item.



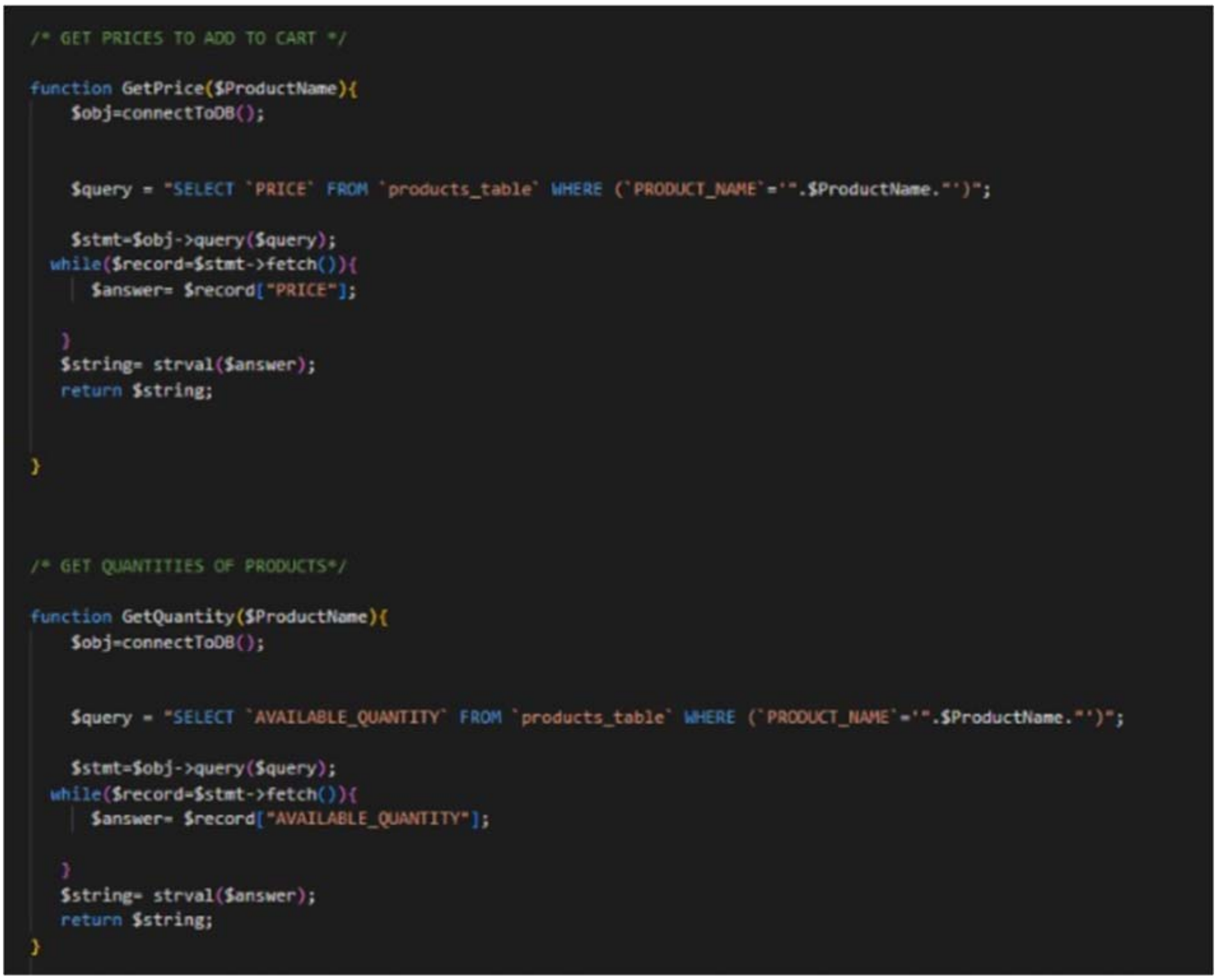
```
//function that will save the number of selected items
function addProduct(name,q,num){
  var x=parseInt(document.getElementById(name).innerHTML)+1;
  //if the selected ones become greater than the quantity found in the system it cancels the last addition
  if(x>parseInt(q)){
    alert("The quantity of this product is finished...");
    document.getElementById(name).innerHTML=x-1;
    cart=Number(cart)-num;
  }
  //if quantity selected is within the available quantity in the system
  else{
    document.getElementById(name).innerHTML=x;
    //open the purchase button in order if any purchase the customer likes to do
    document.getElementById("purchase").style.pointerEvents="visible";
    document.getElementById("label").style.pointerEvents="visible";
    document.getElementById("label").disabled=false;
    document.getElementById("purchase").disabled=false;
    //save the cart value
    updateCart();

  }

}
```

This is the addProduct function which checks whether the last added product doesn't exceed the quantity of the products available. If it does exceed the user will be alerted. Else the visuals will change to highlight the addition and the cart will be updated. The availability of products will be taken from a PHP file described later.



```
//the function that will save the cart value
function updateCart(){
  document.getElementById("cart").innerHTML=cart;
}
```

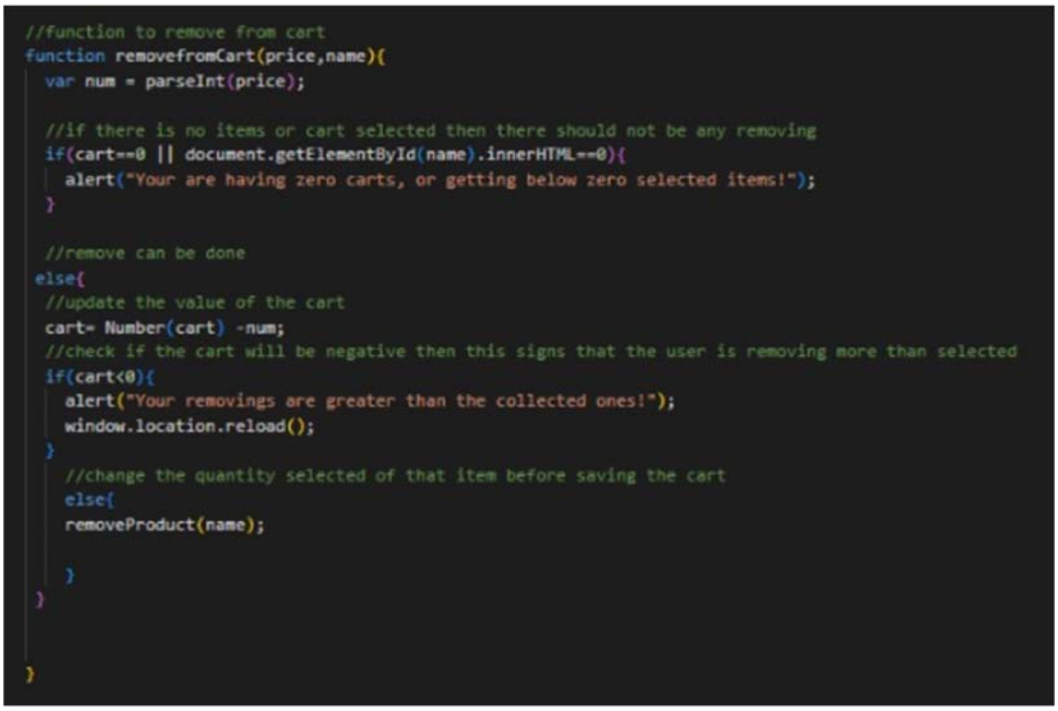
This is just to update the current cart after adding a product.



```
/* GET PRICES TO ADD TO CART */

function GetPrice($ProductName){
    $obj=connectToDB();


    $query = "SELECT `PRICE` FROM `products_table` WHERE (`PRODUCT_NAME`='".$ProductName."')";

    $stmt=$obj->query($query);
  while($record=$stmt->fetch()){
      $answer= $record["PRICE"];

   }
   $string= strval($answer);
   return $string;


}



/* GET QUANTITIES OF PRODUCTS*/

function GetQuantity($ProductName){
    $obj=connectToDB();


    $query = "SELECT `AVAILABLE_QUANTITY` FROM `products_table` WHERE (`PRODUCT_NAME`='".$ProductName."')";

    $stmt=$obj->query($query);
  while($record=$stmt->fetch()){
      $answer= $record["AVAILABLE_QUANTITY"];

   }
   $string= strval($answer);
   return $string;
}
```

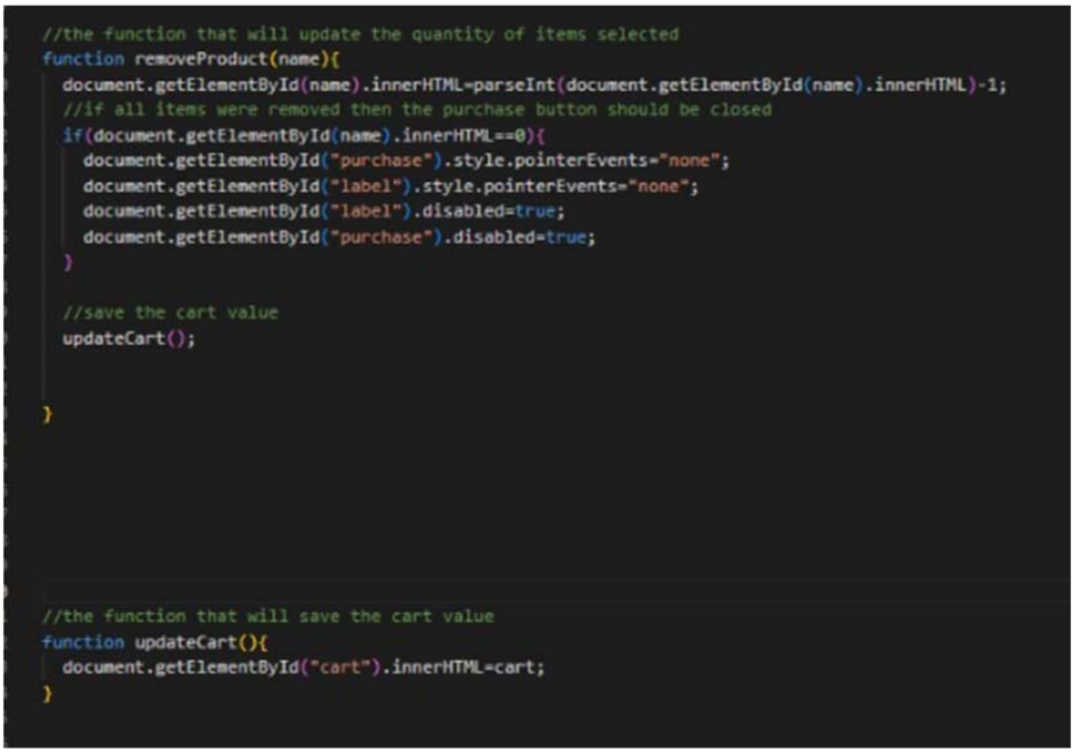
The first method in this PHP file gets searches the database for the product the user wants and returns its price while the second method gets the quantity to ensure that the user can still buy more of this product.

As for the removal from cart:



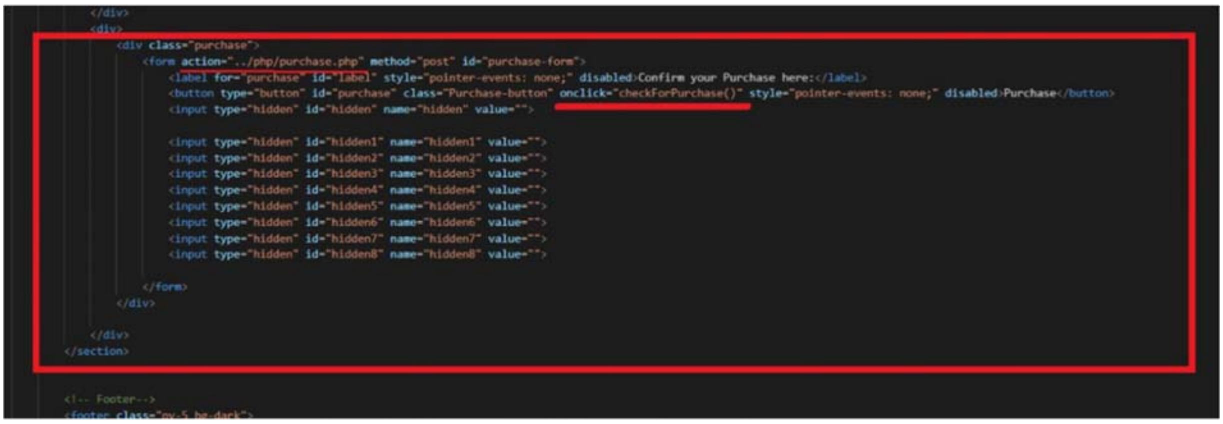
```
//function to remove from cart
function removefromCart(price,name){
  var num = parseInt(price);

  //if there is no items or cart selected then there should not be any removing
  if(cart==0 || document.getElementById(name).innerHTML==0){
    alert("Your are having zero carts, or getting below zero selected items!");
  }

  //remove can be done
 else{
  //update the value of the cart
  cart= Number(cart) -num;
  //check if the cart will be negative then this signs that the user is removing more than selected
  if(cart<0){
    alert("Your removings are greater than the collected ones!");
    window.location.reload();
  }
    //change the quantity selected of that item before saving the cart
    else{
    removeProduct(name);

    }
 }


}
```

If the user presses remove from cart, but he doesn’t have any items or a cart an alert message will be shown. Else, the prices of the cart will be updated. If the cart reaches a negative value, the user will be alerted as this means removing more than the selected items. Otherwise, the item will be removed. The function is as follows:



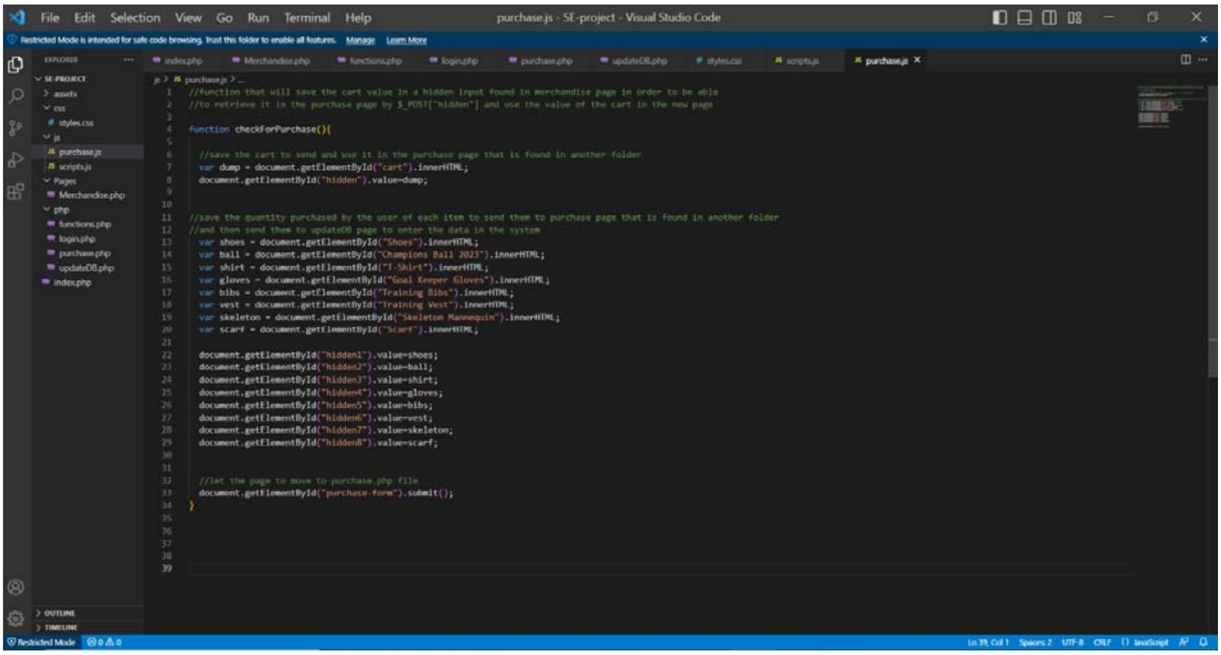
```
//the function that will update the quantity of items selected
function removeProduct(name){
  document.getElementById(name).innerHTML=parseInt(document.getElementById(name).innerHTML)-1;
  //if all items were removed then the purchase button should be closed
  if(document.getElementById(name).innerHTML==0){
    document.getElementById("purchase").style.pointerEvents="none";
    document.getElementById("label").style.pointerEvents="none";
    document.getElementById("label").disabled=true;
    document.getElementById("purchase").disabled=true;
  }

  //save the cart value
  updateCart();


}







//the function that will save the cart value
function updateCart(){
  document.getElementById("cart").innerHTML=cart;
}
```

This changes the style to show that the item was removed and then updates the cart.



```html
        </div>
        <div>
            <div class="purchase">
                <form action="../php/purchase.php" method="post" id="purchase-form">
                    <label for="purchase" id="label" style="pointer-events: none;" disabled>Confirm your Purchase here:</label>
                    <button type="button" id="purchase" class="Purchase-button" onclick="checkForPurchase()" style="pointer-events: none;" disabled>Purchase</button>
                    <input type="hidden" id="hidden" name="hidden" value="">

                    <input type="hidden" id="hidden1" name="hidden1" value="">
                    <input type="hidden" id="hidden2" name="hidden2" value="">
                    <input type="hidden" id="hidden3" name="hidden3" value="">
                    <input type="hidden" id="hidden4" name="hidden4" value="">
                    <input type="hidden" id="hidden5" name="hidden5" value="">
                    <input type="hidden" id="hidden6" name="hidden6" value="">
                    <input type="hidden" id="hidden7" name="hidden7" value="">
                    <input type="hidden" id="hidden8" name="hidden8" value="">

                </form>
            </div>

        </div>
    </section>

    <!-- Footer-->
    <footer class="py-5 bg-dark">
```

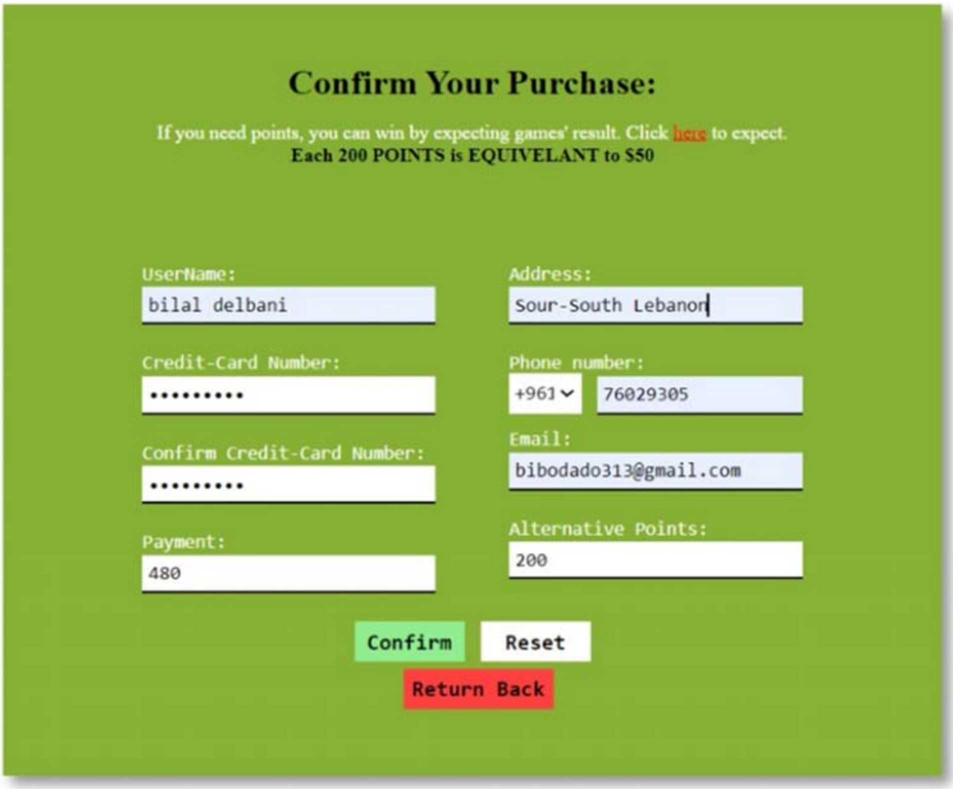
When clicking on the purchase button (which will no longer be disabled after adding items), the action will be handled by the purchase.php and JavaScript method checkForPurchase.



File Edit Selection View Go Run Terminal Help
purchase.js - SE-project - Visual Studio Code
Restricted Mode is intended for safe code browsing. Trust this folder to enable all features. Manage Learn More
EXPLORER
SE-PROJECT
assets
css
styles.css
js
purchase.js
scripts.js
Pages
Merchandise.php
php
functions.php
login.php
purchase.php
updateDB.php
index.php
index.php Merchandise.php functions.php login.php purchase.php updateDB.php styles.css scripts.js purchase.js
js > purchase.js > ...
OUTLINE
TIMELINE
Restricted Mode
Ln 39, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript

```javascript
//function that will save the cart value in a hidden input found in merchandise page in order to be able
//to retrieve it in the purchase page by $_POST["hidden"] and use the value of the cart in the new page

function checkForPurchase(){

  //save the cart to send and use it in the purchase page that is found in another folder
  var dump = document.getElementById("cart").innerHTML;
  document.getElementById("hidden").value=dump;


//save the quantity purchased by the user of each item to send them to purchase page that is found in another folder
//and then send them to updateDB page to enter the data in the system
  var shoes = document.getElementById("Shoes").innerHTML;
  var ball = document.getElementById("Champions Ball 2023").innerHTML;
  var shirt = document.getElementById("T-Shirt").innerHTML;
  var gloves = document.getElementById("Goal Keeper Gloves").innerHTML;
  var bibs = document.getElementById("Training Bibs").innerHTML;
  var vest = document.getElementById("Training Vest").innerHTML;
  var skeleton = document.getElementById("Skeleton Mannequin").innerHTML;
  var scarf = document.getElementById("Scarf").innerHTML;

  document.getElementById("hidden1").value=shoes;
  document.getElementById("hidden2").value=ball;
  document.getElementById("hidden3").value=shirt;
  document.getElementById("hidden4").value=gloves;
  document.getElementById("hidden5").value=bibs;
  document.getElementById("hidden6").value=vest;
  document.getElementById("hidden7").value=skeleton;
  document.getElementById("hidden8").value=scarf;


  //let the page to move to purchase.php file
  document.getElementById("purchase-form").submit();
}
```

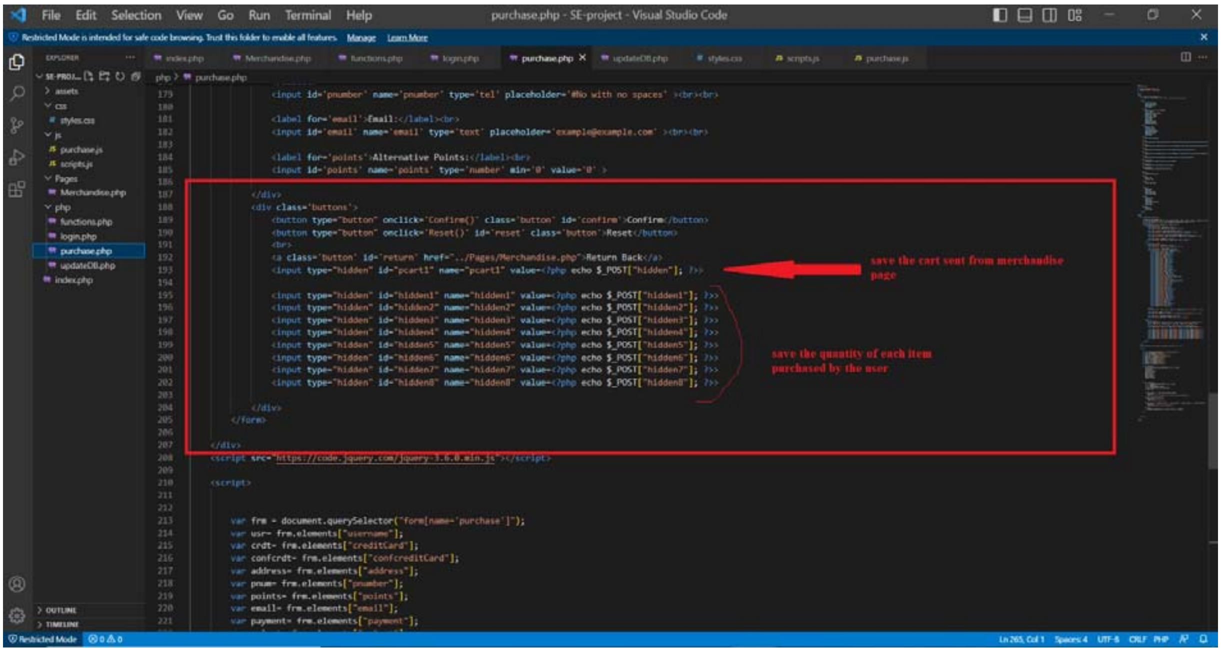
This JavaScript saves the cart purchased by the user. Then it saves the quantity of each item to send to the confirm purchase.php page. Note that this function saves the cart value in a hidden input found on the merchandise page to retrieve it in the purchase page by the $POST.



Confirm Your Purchase:

If you need points, you can win by expecting games' result. Click here to expect.
Each 200 POINTS is EQUIVELANT to $50

UserName: bilal delbani	Address: Sour-South Lebanon
Credit-Card Number: ••••••••••	Phone number: +961 76029305
Confirm Credit-Card Number: ••••••••••	Email: bibodado313@gmail.com
Payment: 480	Alternative Points: 200

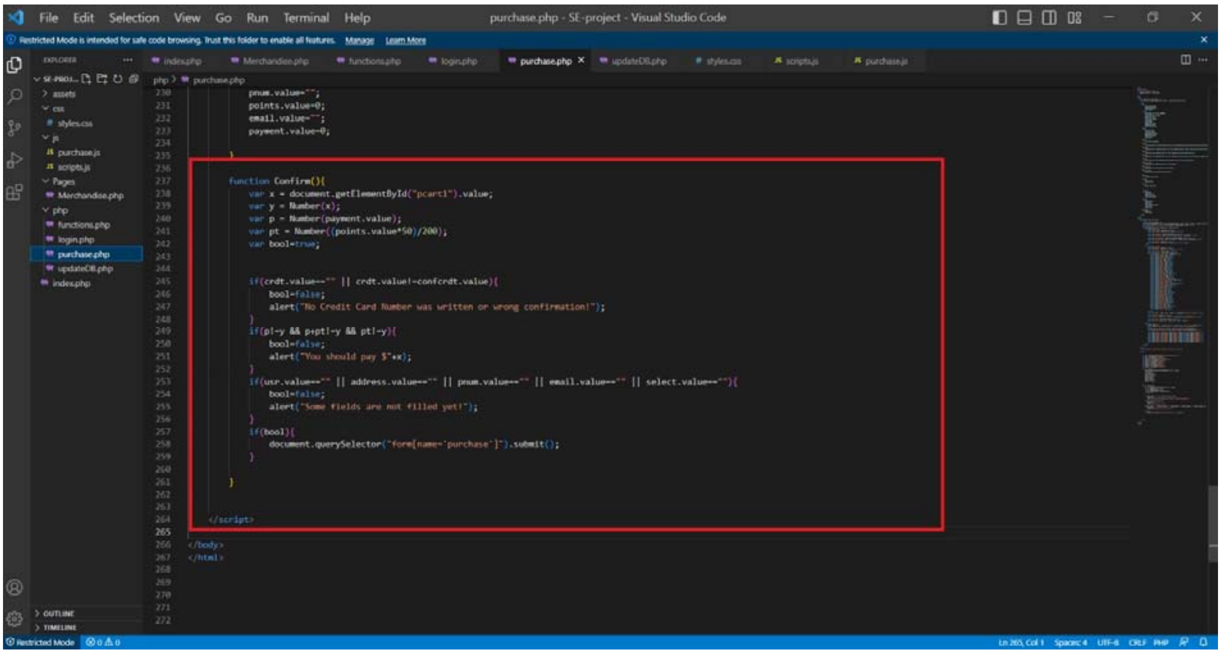


```
                <input id='pnumber' name='pnumber' type='tel' placeholder='#No with no spaces' ><br><br>

                <label for='email'>Email:</label><br>
                <input id='email' name='email' type='text' placeholder='example@example.com' ><br><br>

                <label for='points'>Alternative Points:</label><br>
                <input id='points' name='points' type='number' min='0' value='0' >

            </div>
            <div class='buttons'>
                <button type="button" onclick='Confirm()' class='button' id='confirm'>Confirm</button>
                <button type="button" onclick='Reset()' id='reset' class='button'>Reset</button>
                <br>
                <a class='button' id='return' href="../Pages/Merchandise.php">Return Back</a>
                <input type="hidden" id="pcart1" name="pcart1" value=<?php echo $_POST["hidden"]; ?>>

                <input type="hidden" id="hidden1" name="hidden1" value=<?php echo $_POST["hidden1"]; ?>>
                <input type="hidden" id="hidden2" name="hidden2" value=<?php echo $_POST["hidden2"]; ?>>
                <input type="hidden" id="hidden3" name="hidden3" value=<?php echo $_POST["hidden3"]; ?>>
                <input type="hidden" id="hidden4" name="hidden4" value=<?php echo $_POST["hidden4"]; ?>>
                <input type="hidden" id="hidden5" name="hidden5" value=<?php echo $_POST["hidden5"]; ?>>
                <input type="hidden" id="hidden6" name="hidden6" value=<?php echo $_POST["hidden6"]; ?>>
                <input type="hidden" id="hidden7" name="hidden7" value=<?php echo $_POST["hidden7"]; ?>>
                <input type="hidden" id="hidden8" name="hidden8" value=<?php echo $_POST["hidden8"]; ?>>

            </div>
        </form>

    </div>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script>


        var frm = document.querySelector("form[name='purchase']");
        var usr= frm.elements["username"];
        var crdt= frm.elements["creditCard"];
        var confcrdt= frm.elements["confcreditCard"];
        var address= frm.elements["address"];
        var pnum= frm.elements["pnumber"];
        var points= frm.elements["points"];
        var email= frm.elements["email"];
        var payment= frm.elements["payment"];
```

save the cart sent from merchandise page

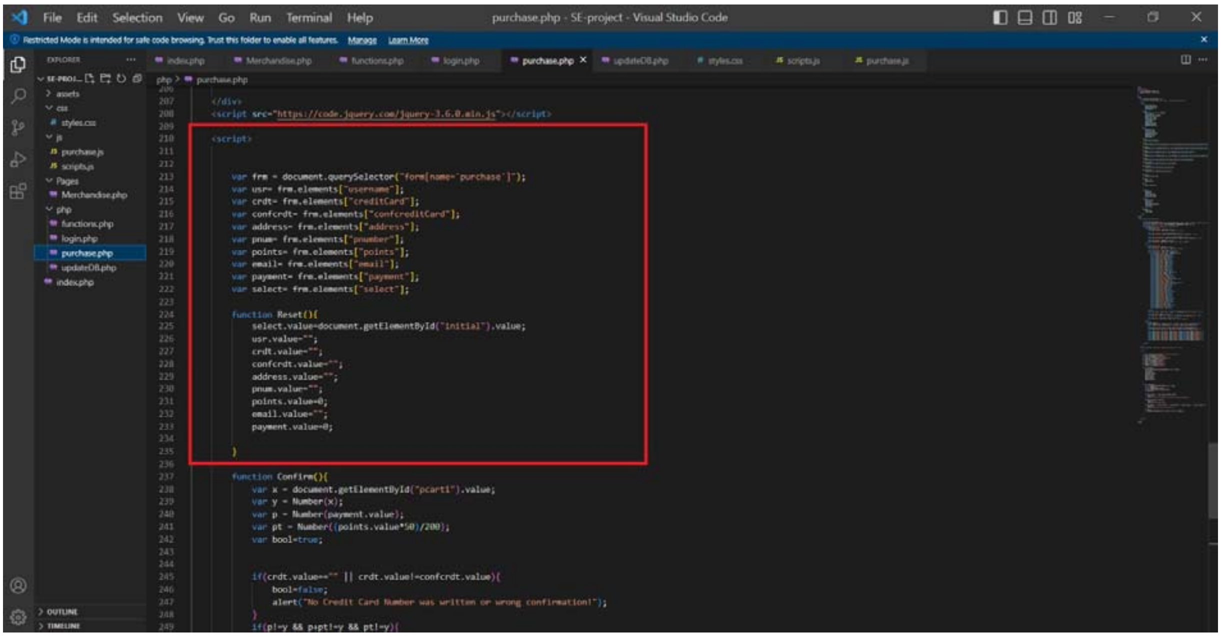
save the quantity of each item purchased by the user

This is the purchase.php file which shows how we save the cart sent from the merchandise page and how we saved the quantity of each item selected by the user.



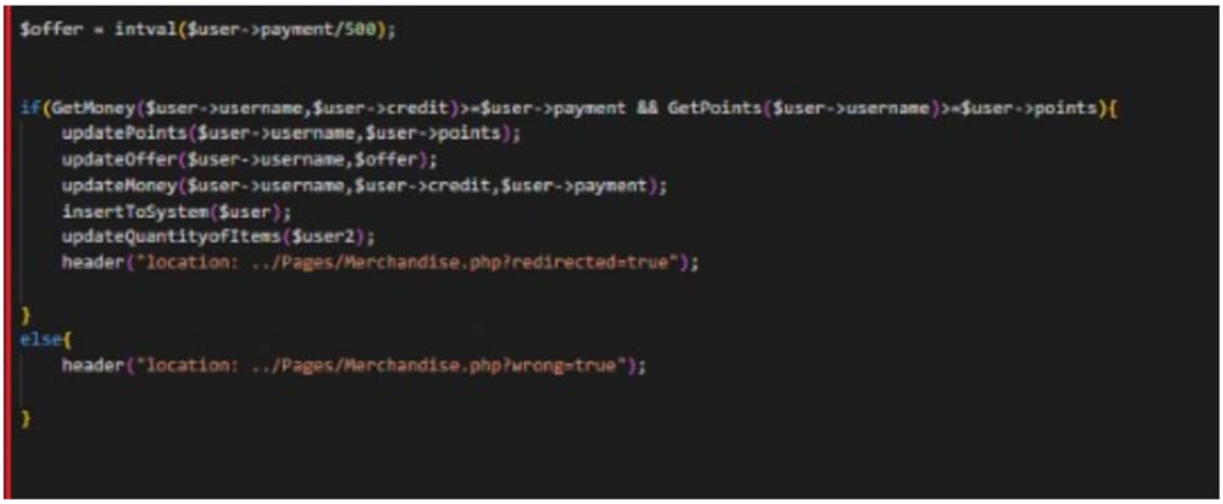
```
            pnum.value="";
            points.value=0;
            email.value="";
            payment.value=0;

        }

        function Confirm(){
            var x = document.getElementById("pcart1").value;
            var y = Number(x);
            var p = Number(payment.value);
            var pt = Number((points.value*50)/200);
            var bool=true;


            if(crdt.value=="" || crdt.value!=confcrdt.value){
                bool=false;
                alert("No Credit Card Number was written or wrong confirmation!");
            }
            if(p!=y && p+pt!=y && pt!=y){
                bool=false;
                alert("You should pay $"+x);
            }
            if(usr.value=="" || address.value=="" || pnum.value=="" || email.value=="" || select.value==""){
                bool=false;
                alert("Some fields are not filled yet!");
            }
            if(bool){
                document.querySelector("form[name='purchase']").submit();
            }

        }



    </script>

</body>
</html>
```

This method in the purchase.php file calculates the amount of money and points the user has (200 points = 50$). Then, it checks that the credit card field is filled, otherwise, the user will be alerted. Whether or not the user has enough money to pay will also be checked and the user will be alerted in case he can't. Also, every field is checked to make sure the user is not missing anything to confirm the purchase.

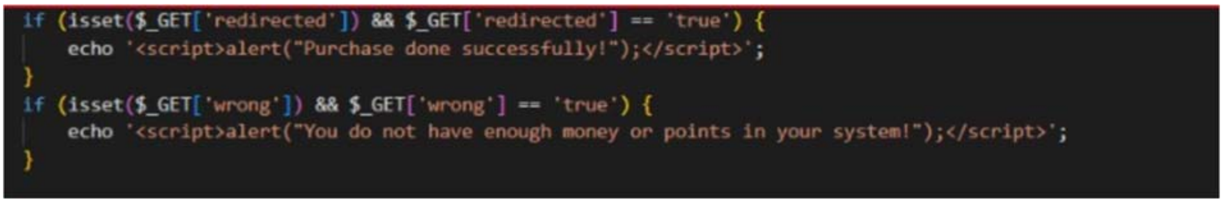


```
    </div>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script>


        var frm = document.querySelector("form[name='purchase']");
        var usr= frm.elements["username"];
        var crdt= frm.elements["creditCard"];
        var confcrdt= frm.elements["confcreditCard"];
        var address= frm.elements["address"];
        var pnum= frm.elements["pnumber"];
        var points= frm.elements["points"];
        var email= frm.elements["email"];
        var payment= frm.elements["payment"];
        var select= frm.elements["select"];

        function Reset(){
            select.value=document.getElementById("initial").value;
            usr.value="";
            crdt.value="";
            confcrdt.value="";
            address.value="";
            pnum.value="";
            points.value=0;
            email.value="";
            payment.value=0;

        }

        function Confirm(){
            var x = document.getElementById("pcart1").value;
            var y = Number(x);
            var p = Number(payment.value);
            var pt = Number((points.value*50)/200);
            var bool=true;


            if(crdt.value=="" || crdt.value!=confcrdt.value){
                bool=false;
                alert("No Credit Card Number was written or wrong confirmation!");
            }
            if(p!=y && p+pt!=y && pt!=y){
```

This is the function for the reset button which takes the value of each form filled by the user and set it to an empty string making it empty.

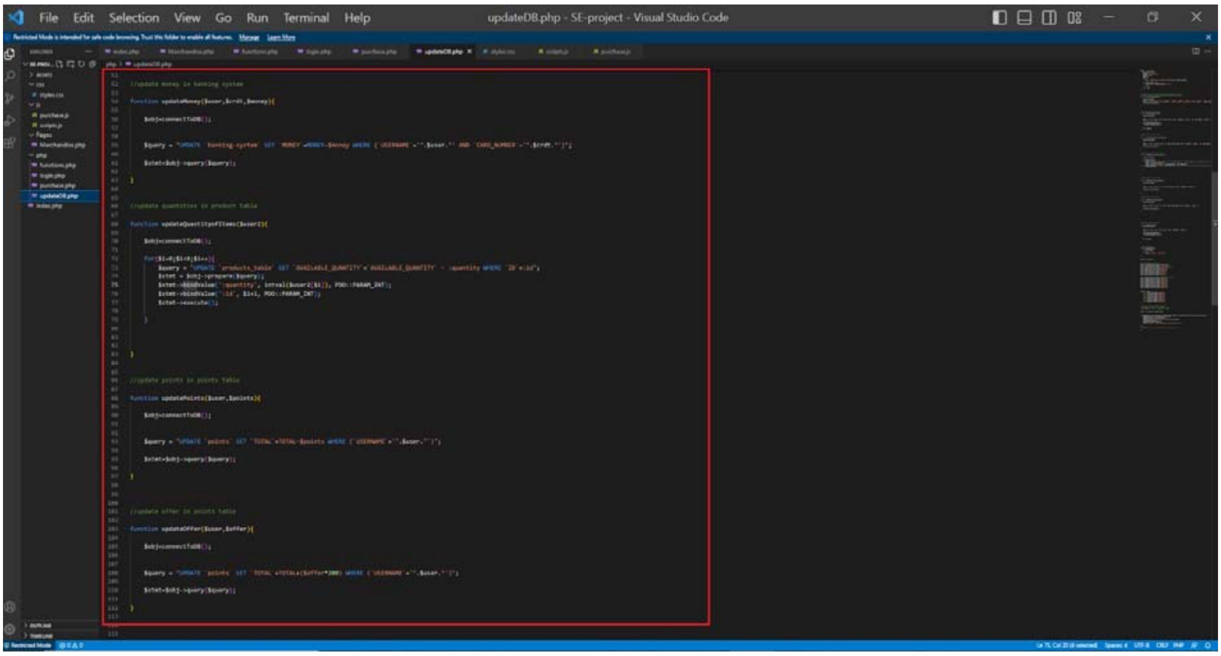


```php
$offer = intval($user->payment/500);


if(GetMoney($user->username,$user->credit)>=$user->payment && GetPoints($user->username)>=$user->points){
    updatePoints($user->username,$user->points);
    updateOffer($user->username,$offer);
    updateMoney($user->username,$user->credit,$user->payment);
    insertToSystem($user);
    updateQuantityofItems($user2);
    header("location: ../Pages/Merchandise.php?redirected=true");

}
else{
    header("location: ../Pages/Merchandise.php?wrong=true");

}
```



```php
if (isset($_GET['redirected']) && $_GET['redirected'] == 'true') {
    echo '<script>alert("Purchase done successfully!");</script>';
}
if (isset($_GET['wrong']) && $_GET['wrong'] == 'true') {
    echo '<script>alert("You do not have enough money or points in your system!");</script>';
}
```

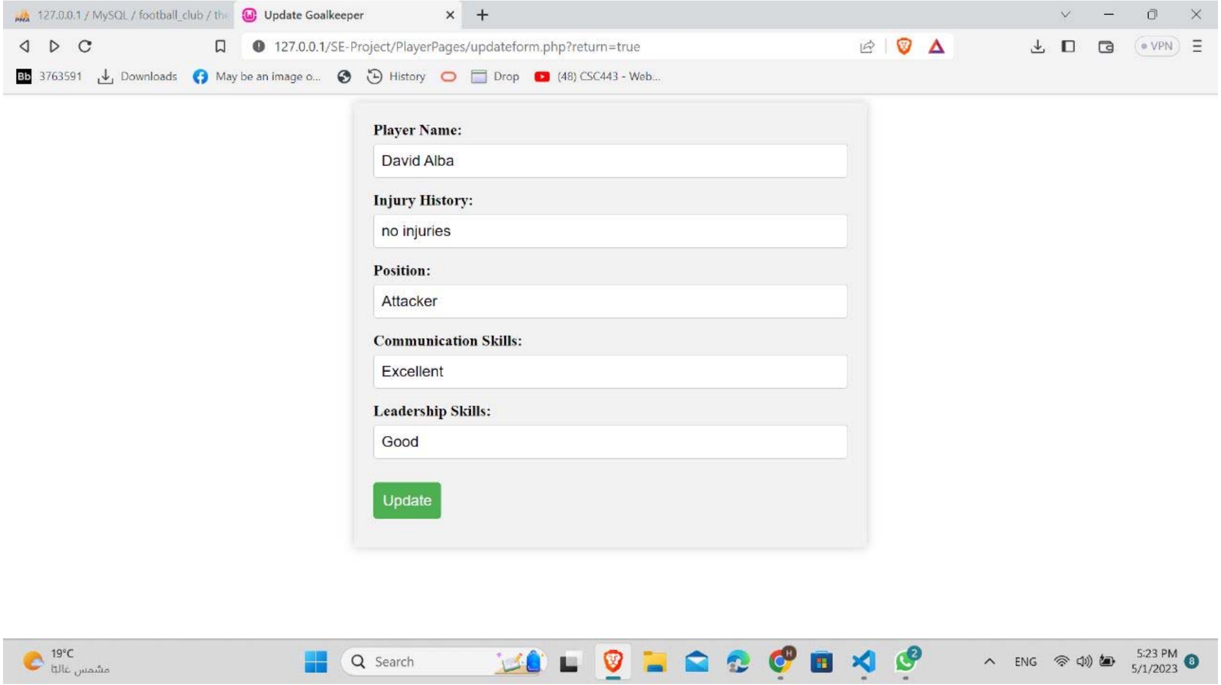
This code checks for the confirmed purchase and updates the points (for every 500 $ spent the user gets 50 points), updates the offer, money, inserts to the system and updates the number of items, and then goes back to the merchandise page. These methods will be explained in detail. The php code checks whether the transaction was successful and whether the user has enough money and alerts accordingly.


File Edit Selection View Go Run Terminal Help
updateDB.php - SE-project - Visual Studio Code

```
//update money in banking system

function updateMoney($user,$crdt,$money){

    $obj=connectToDB();


    $query = "UPDATE `banking-system` SET `MONEY`=MONEY-$money WHERE (`USERNAME`='".$user."' AND `CARD_NUMBER`='".$crdt."')";

    $stmt=$obj->query($query);

}


//update quantities in product table

function updateQuantityofItems($user2){

    $obj=connectToDB();

    for($i=0;$i<8;$i++){
        $query = "UPDATE `products_table` SET `AVAILABLE_QUANTITY`= AVAILABLE_QUANTITY - :quantity WHERE `ID`=:id";
        $stmt = $obj->prepare($query);
        $stmt->bindValue(':quantity', intval($user2[$i]), PDO::PARAM_INT);
        $stmt->bindValue(':id', $i+1, PDO::PARAM_INT);
        $stmt->execute();

    }



}


//update points in points table

function updatePoints($user,$points){

    $obj=connectToDB();


    $query = "UPDATE `points` SET `TOTAL`=TOTAL-$points WHERE (`USERNAME`='".$user."')";

    $stmt=$obj->query($query);

}



//update offer in points table

function updateOffer($user,$offer){

    $obj=connectToDB();


    $query = "UPDATE `points` SET `TOTAL`=TOTAL+($offer*200) WHERE (`USERNAME`='".$user."')";

    $stmt=$obj->query($query);

}
```

The update function connects to the database using $obj = connectToDB(), and performs SQL query to update the money of the user. The second method updates the quantity of the available items in our database after the purchase of the user. Similarly, the third method connects to the database and subtracts the points the user has from the points available in his account. The final method is showing how the offer is updated in the points table.

Updating player information:



Player Name:

David Alba

Injury History:

no injuries

Position:

Attacker

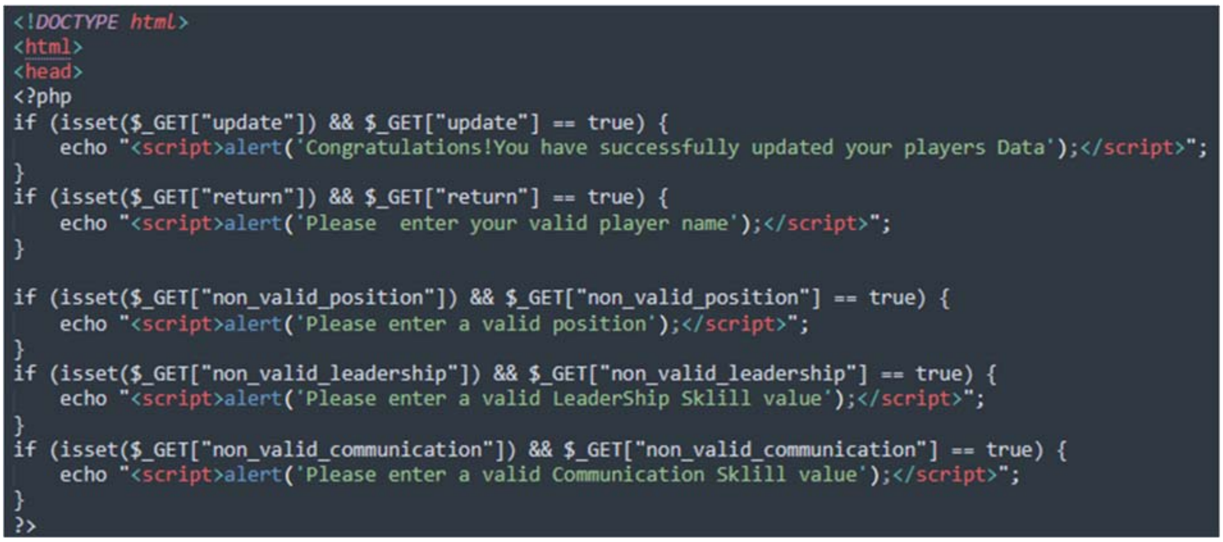
Communication Skills:

Excellent

Leadership Skills:

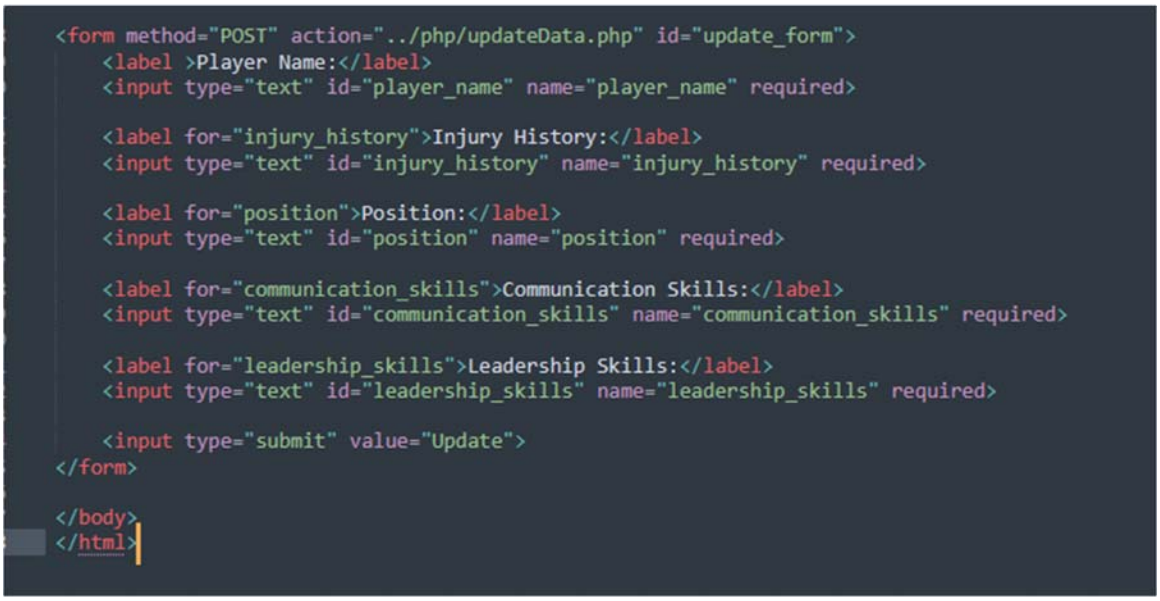
Good

Update



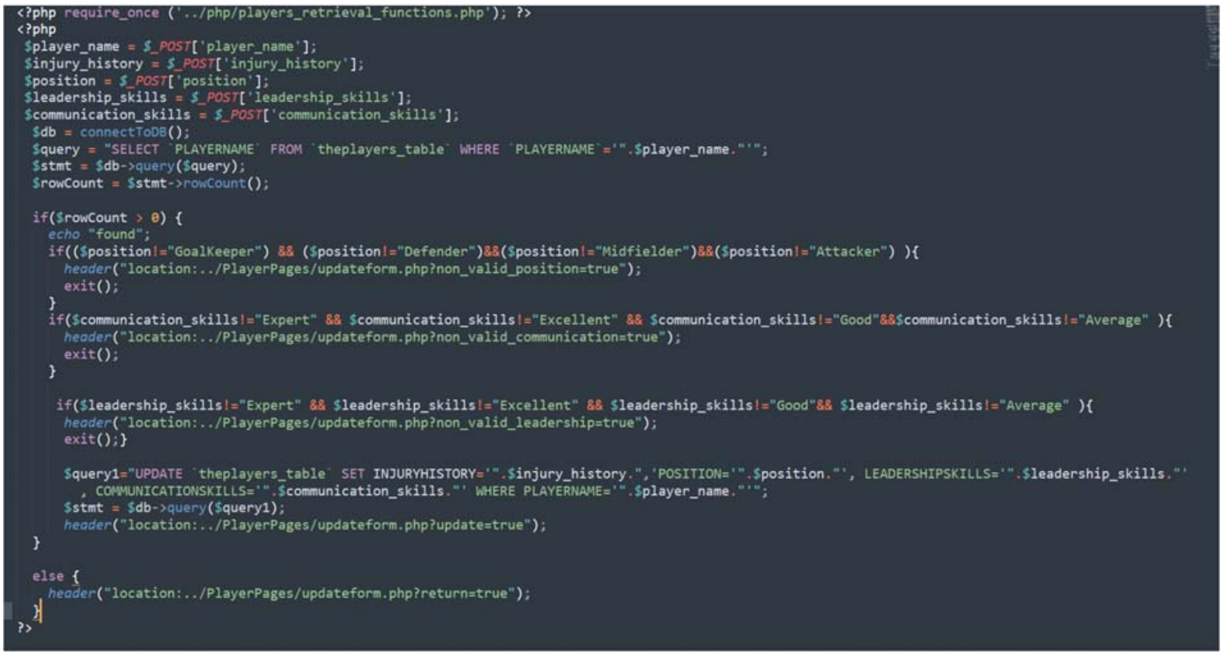
```
<!DOCTYPE html>
<html>
<head>
<?php
if (isset($_GET["update"]) && $_GET["update"] == true) {
    echo "<script>alert('Congratulations!You have successfully updated your players Data');</script>";
}
if (isset($_GET["return"]) && $_GET["return"] == true) {
    echo "<script>alert('Please  enter your valid player name');</script>";
}

if (isset($_GET["non_valid_position"]) && $_GET["non_valid_position"] == true) {
    echo "<script>alert('Please enter a valid position');</script>";
}
if (isset($_GET["non_valid_leadership"]) && $_GET["non_valid_leadership"] == true) {
    echo "<script>alert('Please enter a valid LeaderShip Sklill value');</script>";
}
if (isset($_GET["non_valid_communication"]) && $_GET["non_valid_communication"] == true) {
    echo "<script>alert('Please enter a valid Communication Sklill value');</script>";
}
?>
```

This code ensures that every form entered is valid. If everything is successful, the user will be notified. However, if the player's name, position, leadership description, or communication skills is invalid, the manager will be alerted accordingly.



```
    <form method="POST" action="../php/updateData.php" id="update_form">
        <label >Player Name:</label>
        <input type="text" id="player_name" name="player_name" required>

        <label for="injury_history">Injury History:</label>
        <input type="text" id="injury_history" name="injury_history" required>

        <label for="position">Position:</label>
        <input type="text" id="position" name="position" required>

        <label for="communication_skills">Communication Skills:</label>
        <input type="text" id="communication_skills" name="communication_skills" required>

        <label for="leadership_skills">Leadership Skills:</label>
        <input type="text" id="leadership_skills" name="leadership_skills" required>

        <input type="submit" value="Update">
    </form>

    </body>
    </html>
```

This HTML is for the form that is shown above for the manager to update the player's information.



```php
<?php require_once ('../php/players_retrieval_functions.php'); ?>
<?php
 $player_name = $_POST['player_name'];
 $injury_history = $_POST['injury_history'];
 $position = $_POST['position'];
 $leadership_skills = $_POST['leadership_skills'];
 $communication_skills = $_POST['communication_skills'];
  $db = connectToDB();
  $query = "SELECT `PLAYERNAME` FROM `theplayers_table` WHERE `PLAYERNAME`='".$player_name."'";
  $stmt = $db->query($query);
  $rowCount = $stmt->rowCount();

  if($rowCount > 0) {
    echo "found";
    if(($position!="GoalKeeper") && ($position!="Defender")&&($position!="Midfielder")&&($position!="Attacker") ){
      header("location:../PlayerPages/updateform.php?non_valid_position=true");
      exit();
    }
    if($communication_skills!="Expert" && $communication_skills!="Excellent" && $communication_skills!="Good"&&$communication_skills!="Average" ){
      header("location:../PlayerPages/updateform.php?non_valid_communication=true");
      exit();
    }

     if($leadership_skills!="Expert" && $leadership_skills!="Excellent" && $leadership_skills!="Good"&& $leadership_skills!="Average" ){
      header("location:../PlayerPages/updateform.php?non_valid_leadership=true");
      exit();}

      $query1="UPDATE `theplayers_table` SET INJURYHISTORY='".$injury_history.",'POSITION='".$position."', LEADERSHIPSKILLS='".$leadership_skills."'
        , COMMUNICATIONSKILLS='".$communication_skills."' WHERE PLAYERNAME='".$player_name."'";
      $stmt = $db->query($query1);
      header("location:../PlayerPages/updateform.php?update=true");
  }

  else {
    header("location:../PlayerPages/updateform.php?return=true");
  }
?>
```

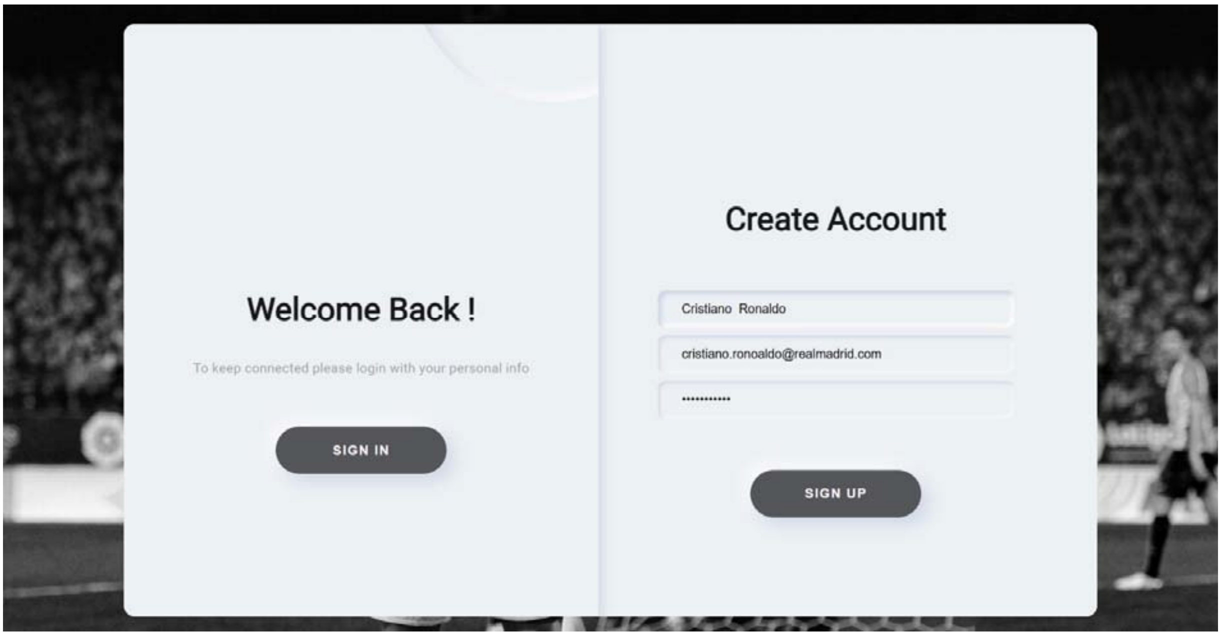
This PHP code sends the result from the queries to find and set the new player's attributes. It connects to the database and queries for the player's name. If rowCount>0 means that there is a record of this player in the database. If the position is not among the valid positions (GoalKeeper, Defender, etc..), the manager will be alerted. The same thing happens with communication and leadership skills. If everything is valid the manager will update successfully. If the rowCount is not positive the manager will be alerted that there is no player in the system's record.

Unit Testing:

Sign up:

1. Successful sign-up
2. Email already exists (Unsuccessful)

1.



Welcome Back !

To keep connected please login with your personal info

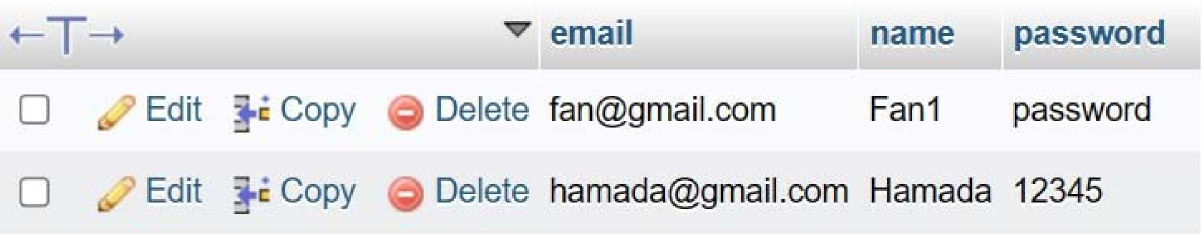
Create Account

Cristiano Ronaldo

cristiano.ronoaldo@realmadrid.com

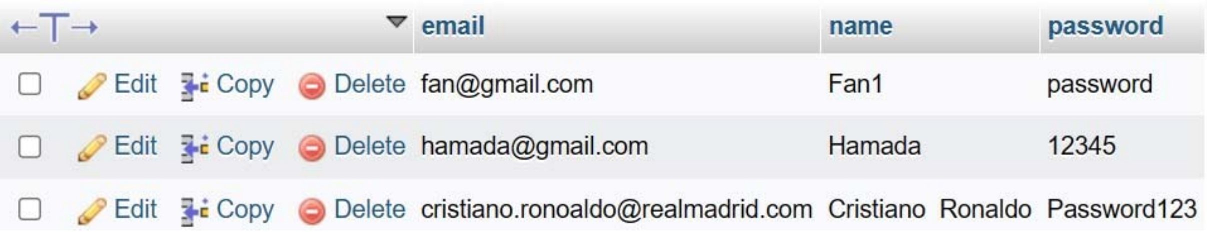
....

Before:



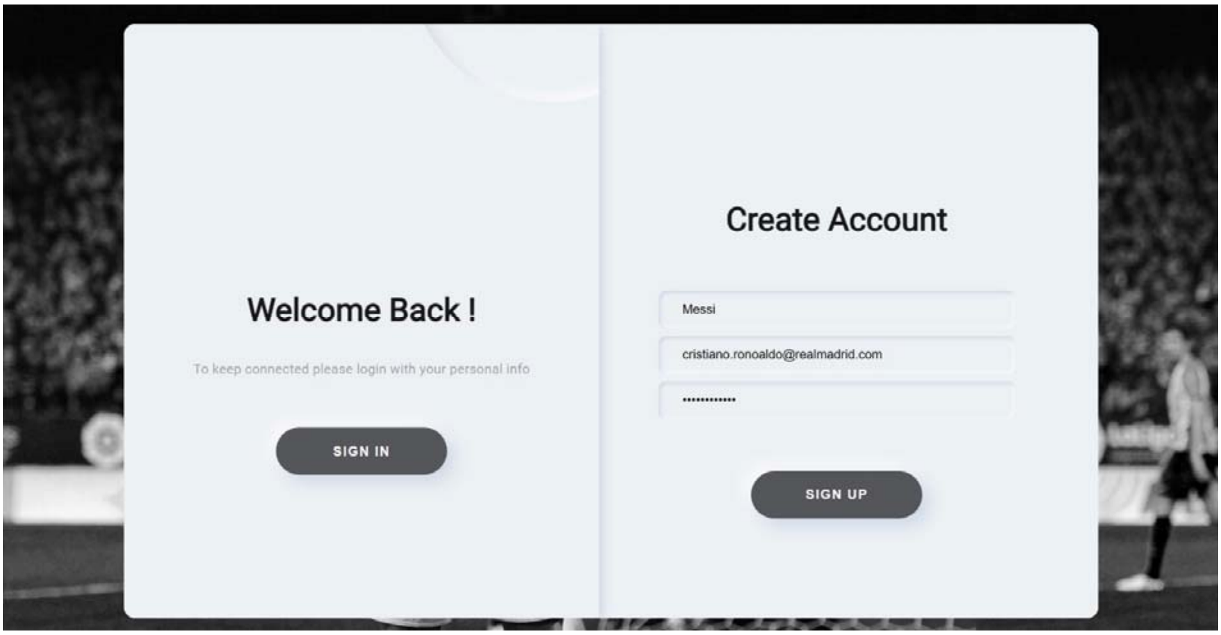
←T→		▼	email	name	password
□	Edit	Copy	Delete	fan@gmail.com	Fan1	password
□	Edit	Copy	Delete	hamada@gmail.com	Hamada	12345

After:



←T→		▼	email	name	password
□	Edit	Copy	Delete	fan@gmail.com	Fan1	password
□	Edit	Copy	Delete	hamada@gmail.com	Hamada	12345
□	Edit	Copy	Delete	cristiano.ronoaldo@realmadrid.com	Cristiano Ronaldo	Password123

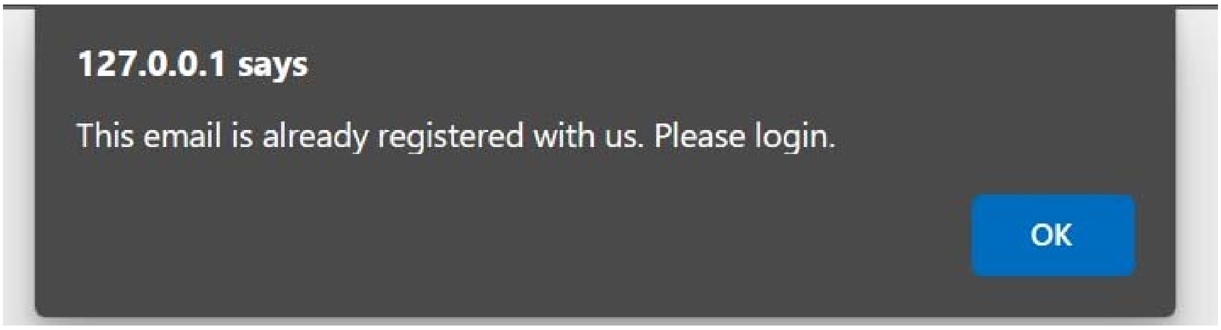
2.



Messi

cristiano.ronoaldo@realmadrid.com

•••••••••••


127.0.0.1 says

This email is already registered with us. Please login.

Test results:

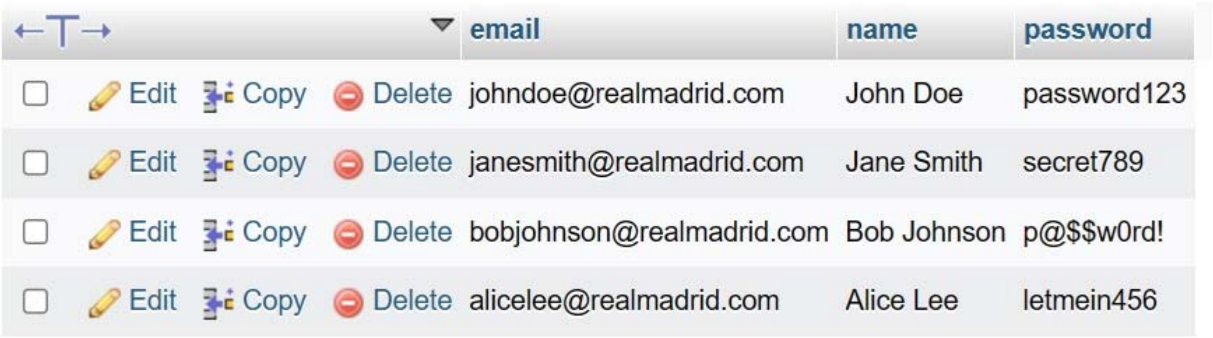
If the email is not already in use successful sign-up and data are added to the database.

If the email is in use the form is not submitted and an alert is sent for the user to change his email.

Logging in:

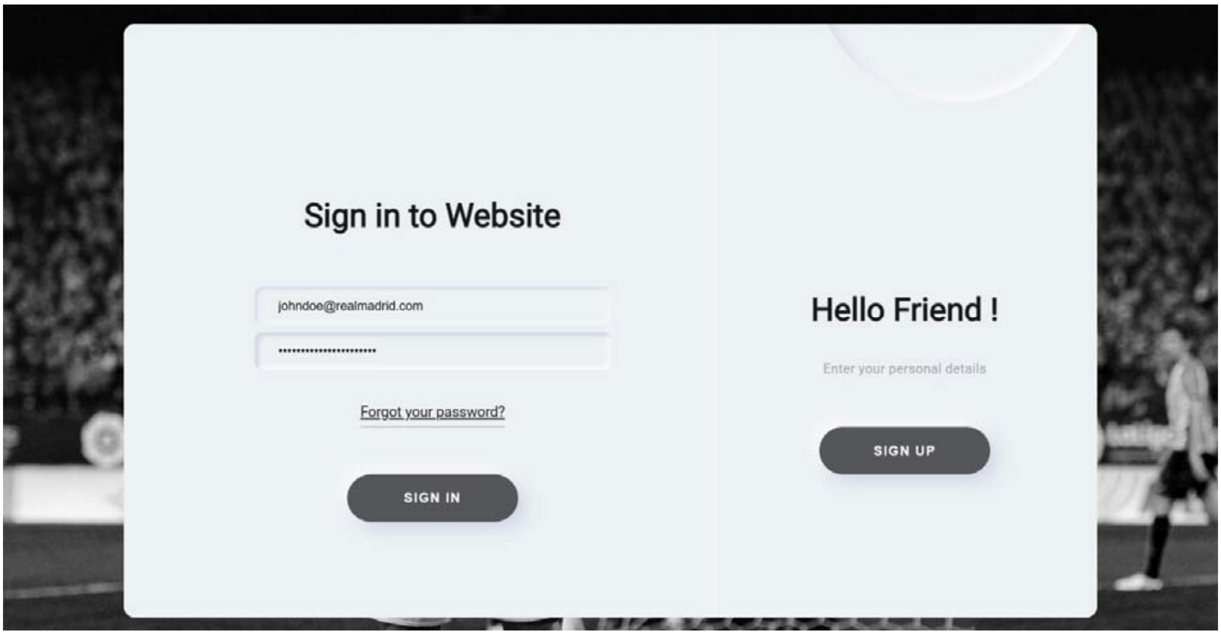
1. Successful login
2. Invalid email or username

Current database:



←T→			▼	email	name	password
□	Edit	Copy	Delete	johndoe@realmadrid.com	John Doe	password123
□	Edit	Copy	Delete	janesmith@realmadrid.com	Jane Smith	secret789
□	Edit	Copy	Delete	bobjohnson@realmadrid.com	Bob Johnson	p@$$w0rd!
□	Edit	Copy	Delete	alicelee@realmadrid.com	Alice Lee	letmein456

1.



Sign in to Website

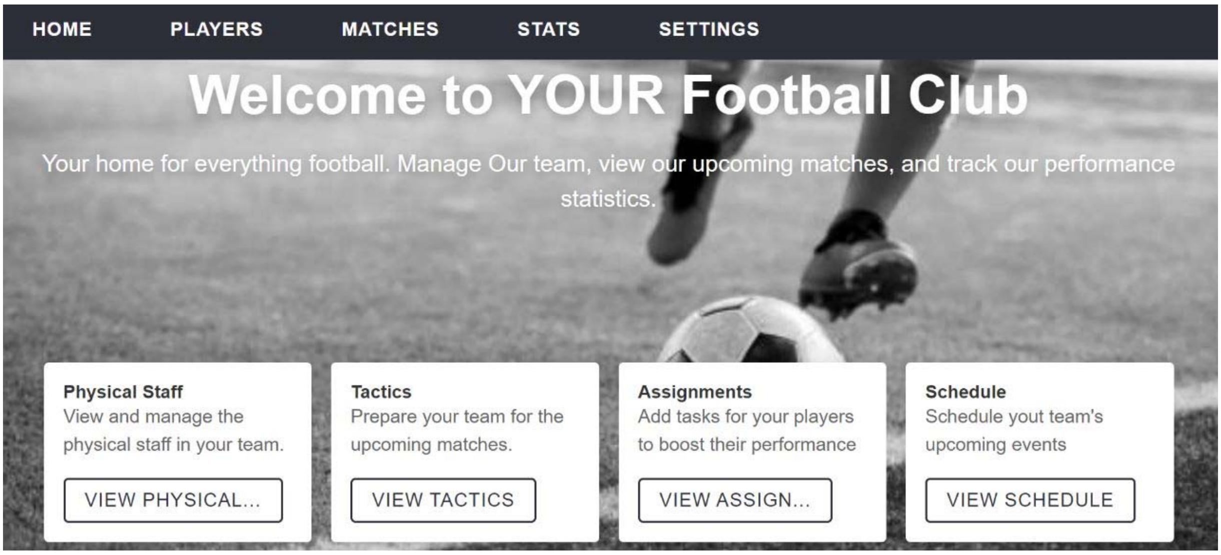
johndoe@realmadrid.com

••••••••••••••••••••••

Forgot your password?

Hello Friend !

Enter your personal details



Welcome to YOUR Football Club

Your home for everything football. Manage Our team, view our upcoming matches, and track our performance statistics.

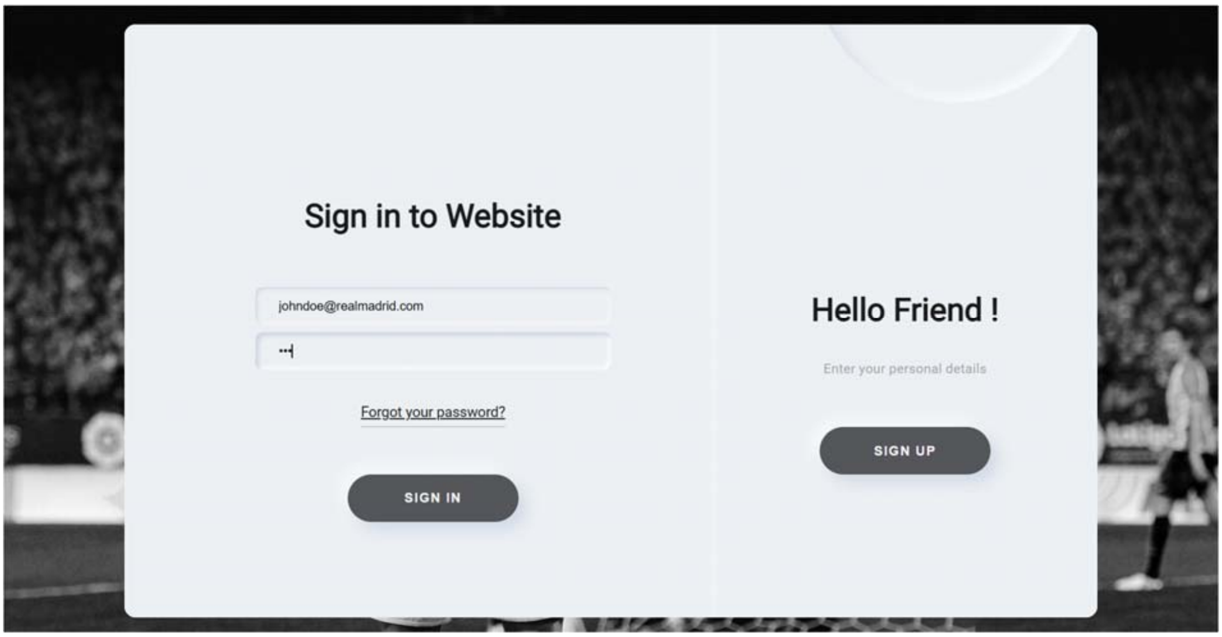
Physical Staff
View and manage the physical staff in your team.

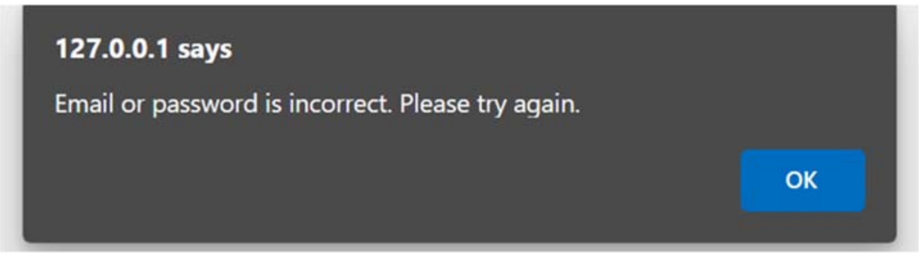
Tactics
Prepare your team for the upcoming matches.

Assignments
Add tasks for your players to boost their performance

Schedule
Schedule yout team's upcoming events

3. Invalid username or password:





127.0.0.1 says

Email or password is incorrect. Please try again.

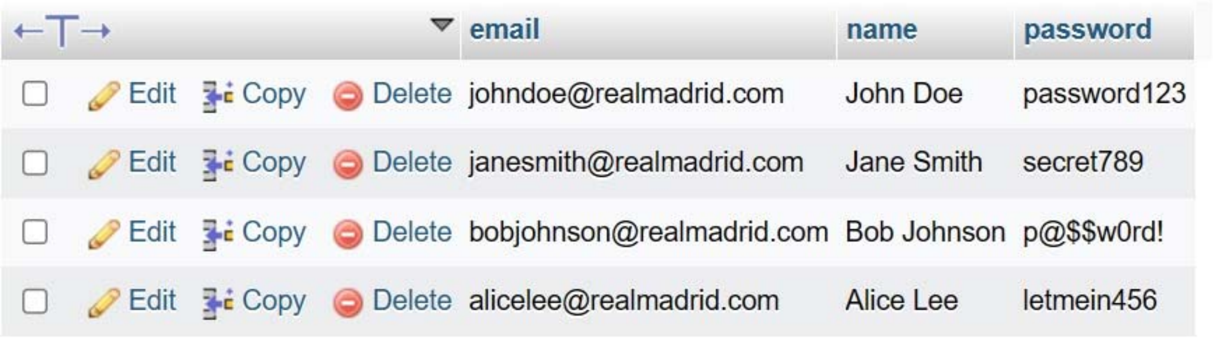
Test results:

When the email and password were valid, the user was able to successfully log in and view the other page. When the email or password was invalid, an alert was sent for the user to try again.

Forgot Password:

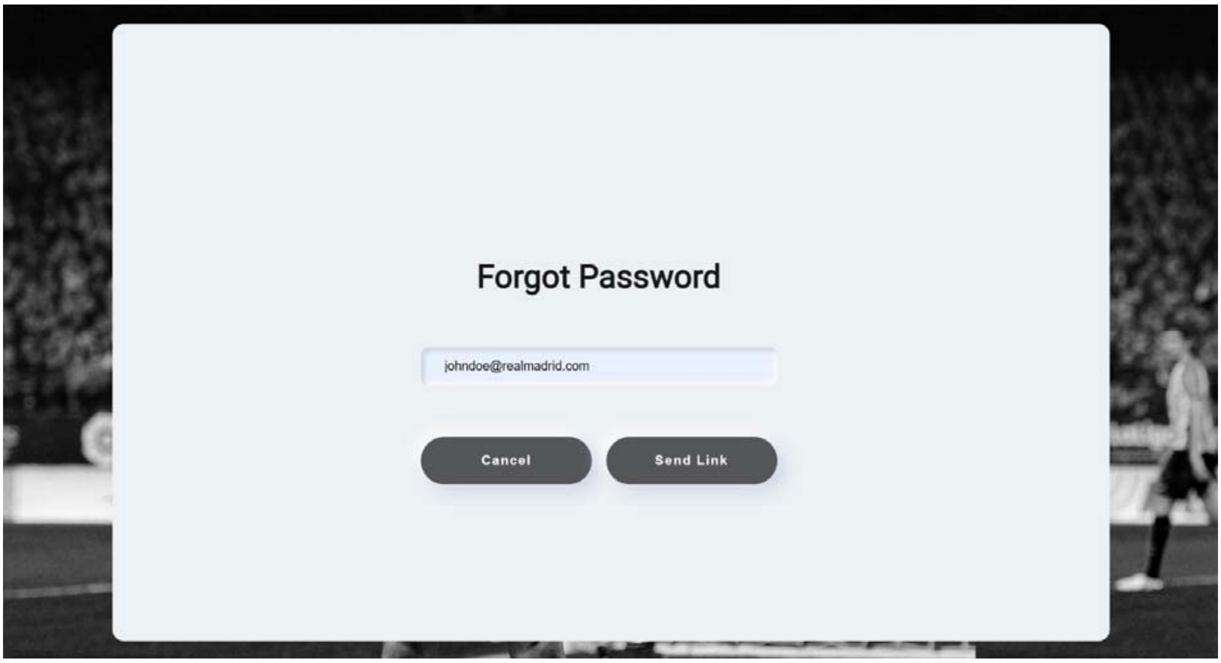
1. Email exists
2. Email doesn't exist

Current database:



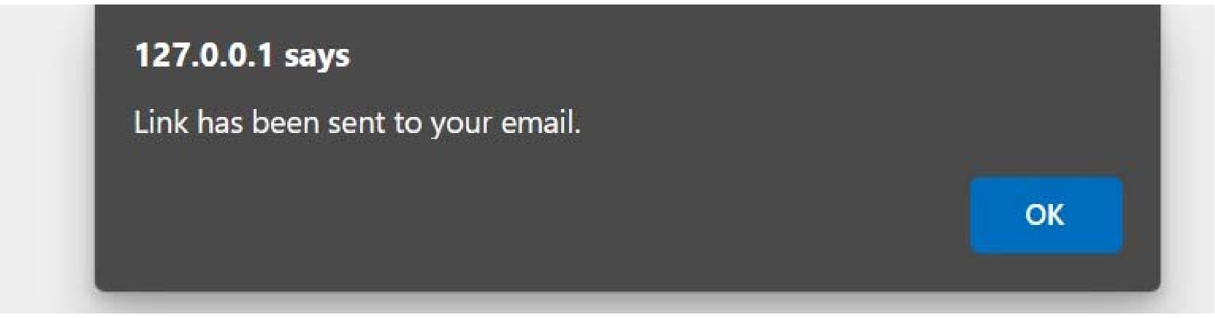
←T→				email	name	password
□	Edit	Copy	Delete	johndoe@realmadrid.com	John Doe	password123
□	Edit	Copy	Delete	janesmith@realmadrid.com	Jane Smith	secret789
□	Edit	Copy	Delete	bobjohnson@realmadrid.com	Bob Johnson	p@$$w0rd!
□	Edit	Copy	Delete	alicelee@realmadrid.com	Alice Lee	letmein456

1.



Forgot Password

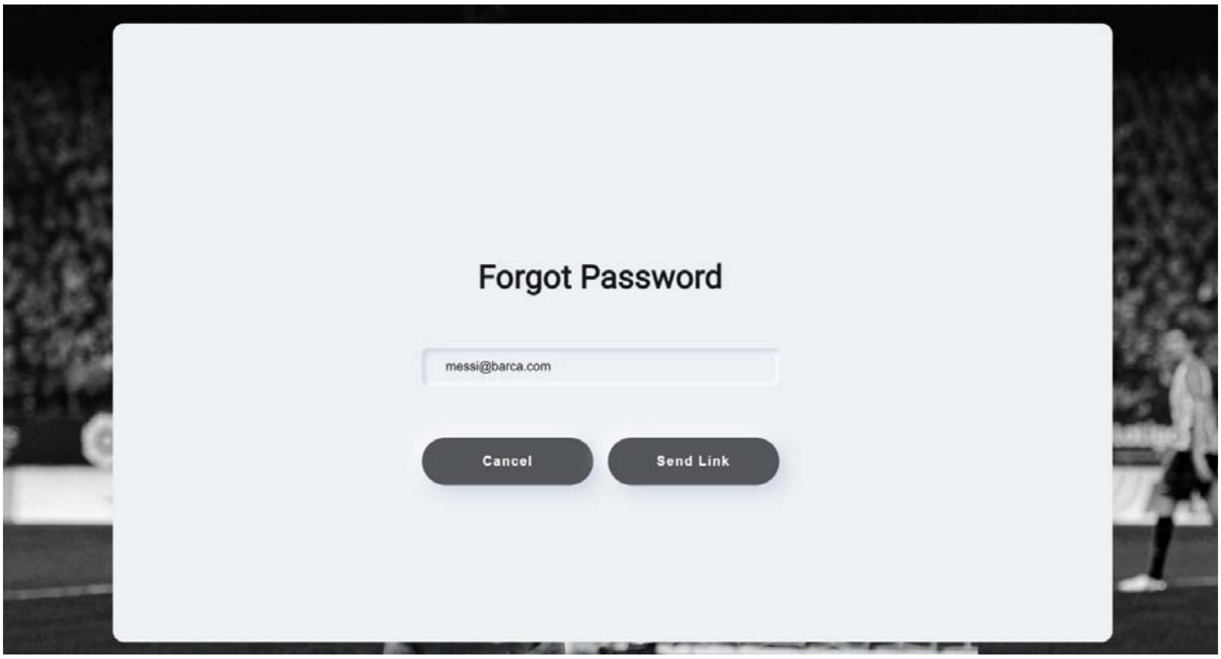
johndoe@realmadrid.com

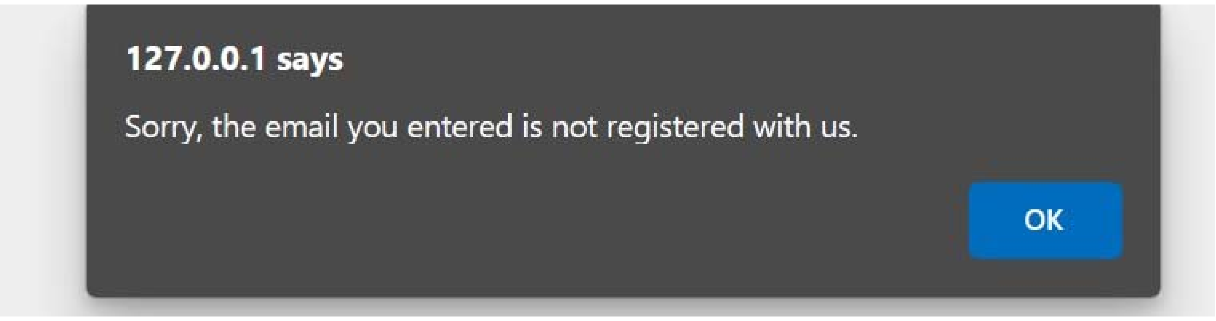


127.0.0.1 says

Link has been sent to your email.

2.




127.0.0.1 says

Sorry, the email you entered is not registered with us.

Test results:

When the email was found in the database, the link was sent to the user. Otherwise, an alert showing that the email is not registered was shown to inform the user.

3. Purchasing Merchandise:

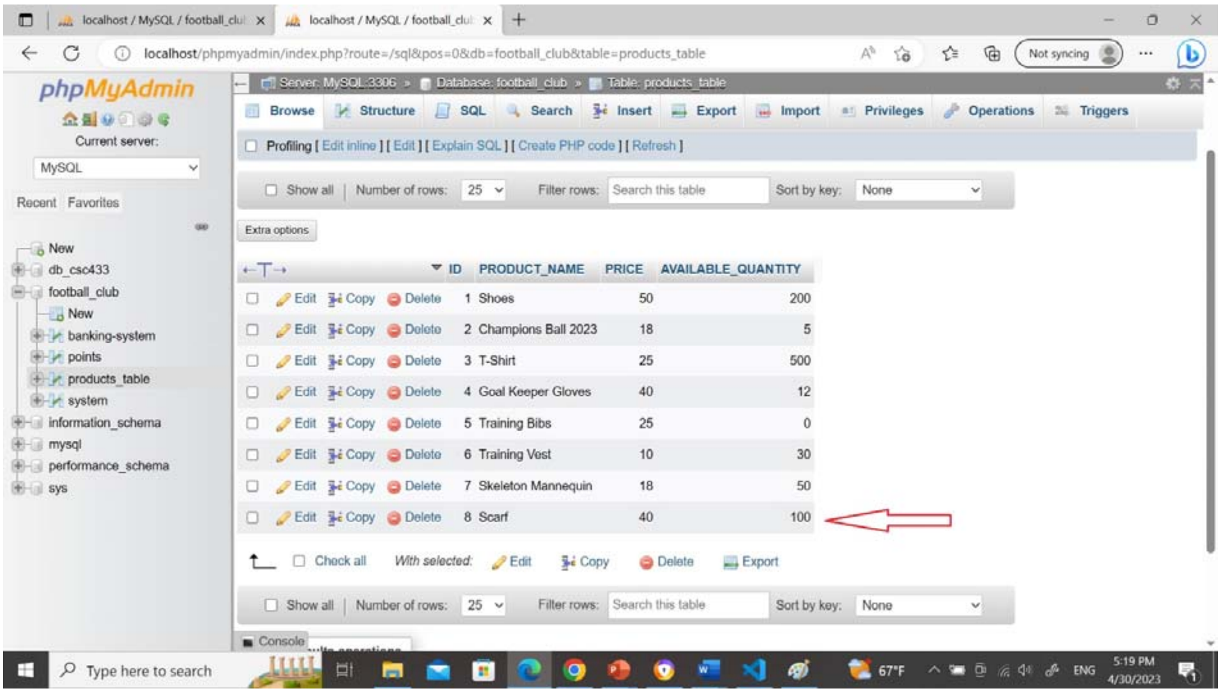
1. Disabled purchase button when no items are selected.
2. Purchase button available when items available in the database are selected.
3. Alert the user when removing the final item from his cart.
4. Unsuccessful purchase (item not available in the database).
5. Fields missing on the purchase page.
6. Credit card information missing.
7. User pays less than the amount he should.

8. Reset button check.

9. Successful purchase.

10. User doesn’t have enough money to pay.

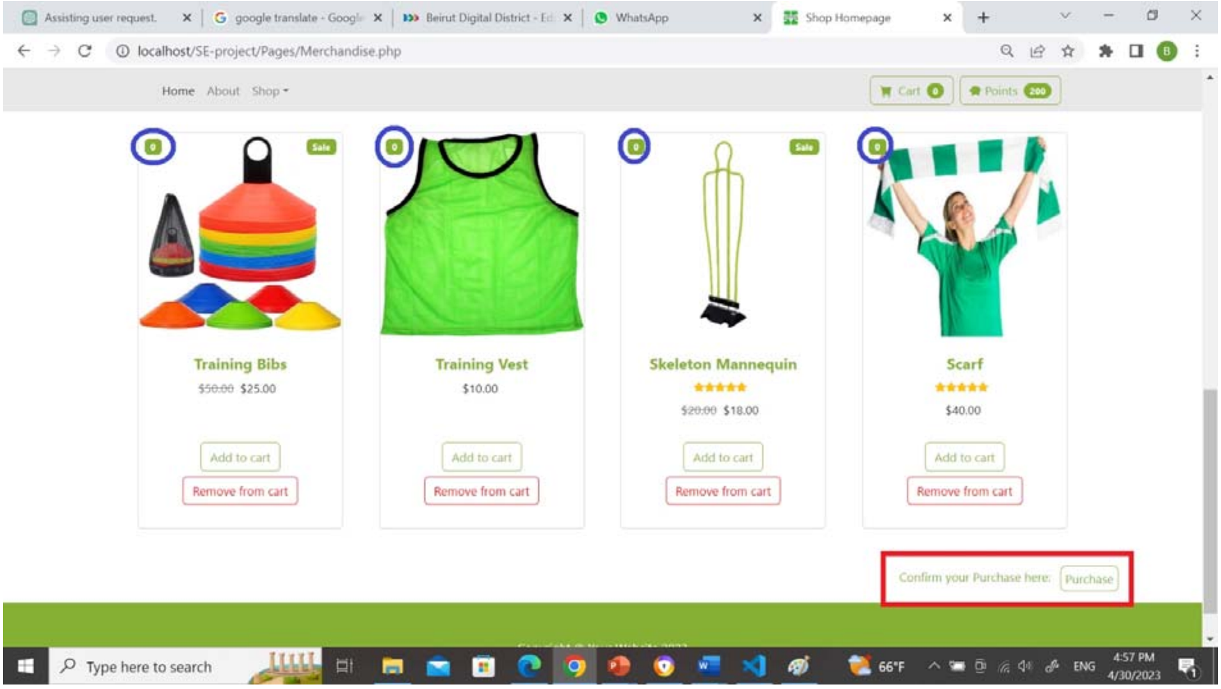
1. Database:



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=football_club&table=products_table

Server: MySQL:3306 » Database: football_club » Table: products_table

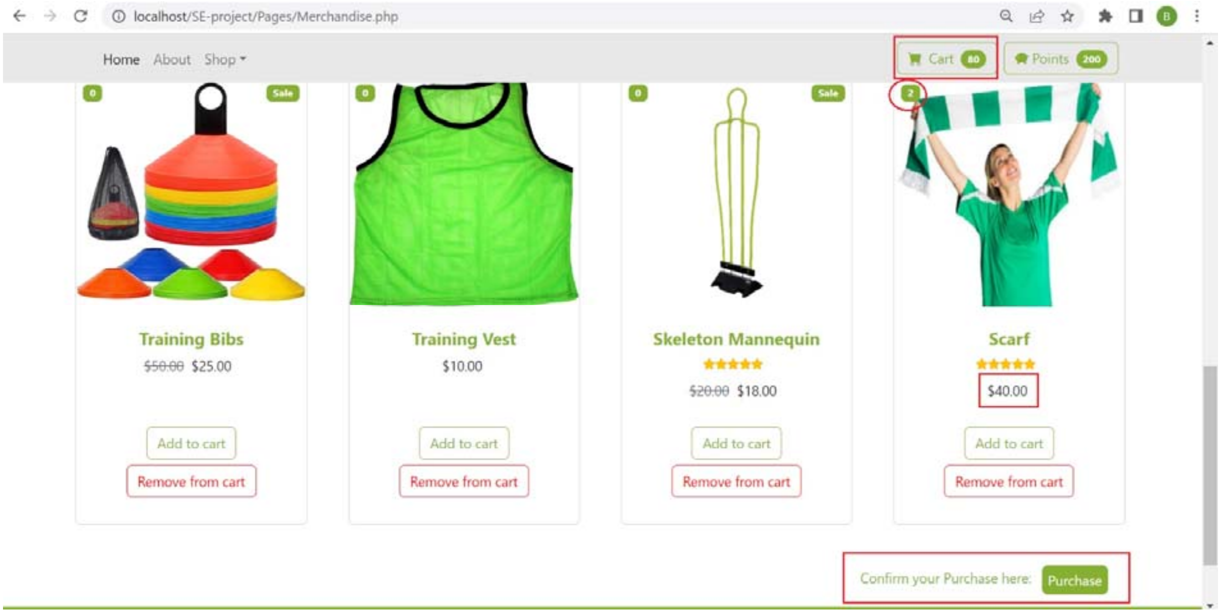
ID	PRODUCT_NAME	PRICE	AVAILABLE_QUANTITY
1	Shoes	50	200
2	Champions Ball 2023	18	5
3	T-Shirt	25	500
4	Goal Keeper Gloves	40	12
5	Training Bibs	25	0
6	Training Vest	10	30
7	Skeleton Mannequin	18	50
8	Scarf	40	100



localhost/SE-project/Pages/Merchandise.php

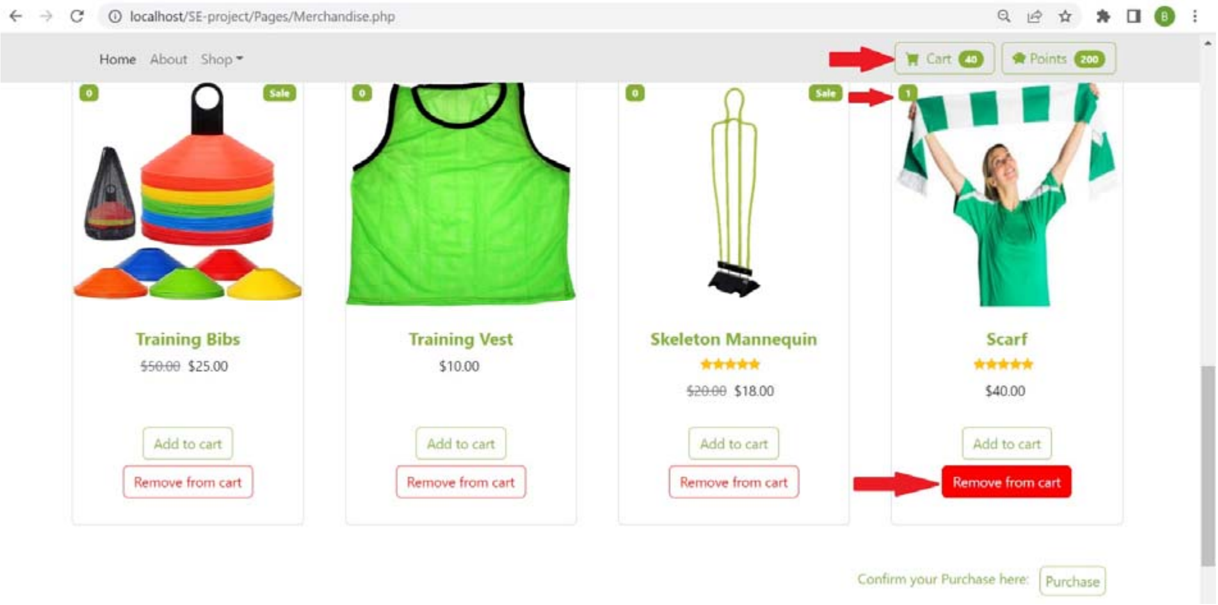
When no items are selected the purchase button is disabled.

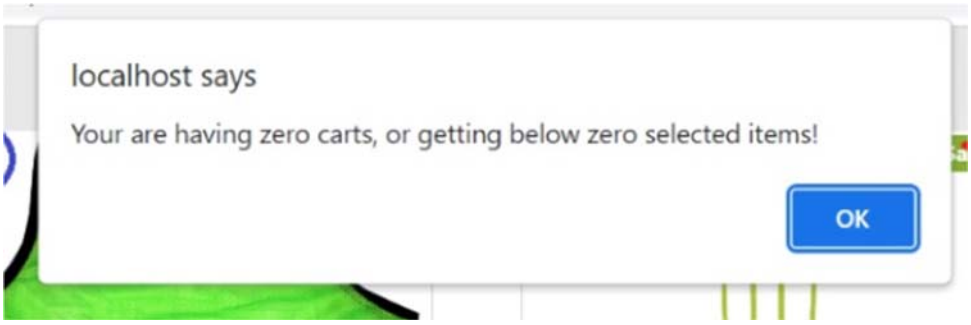
2.



When we selected items available in the database, the purchase button becomes available.

3.



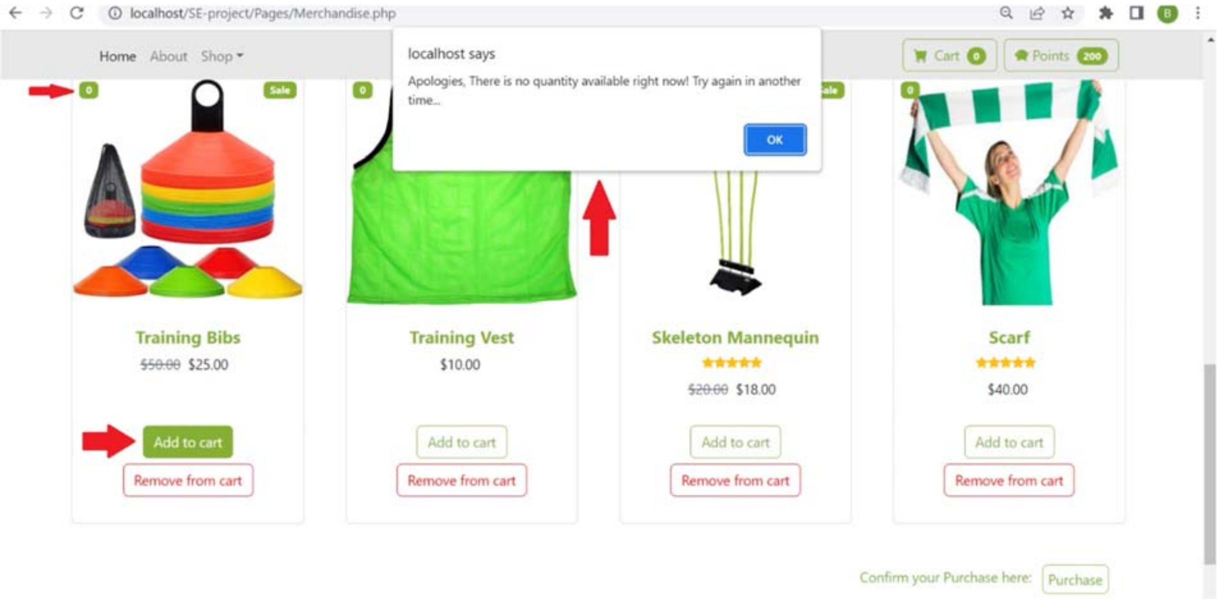


localhost says

Your are having zero carts, or getting below zero selected items!

If we remove the final product selected an alert will be sent to the user.

4.

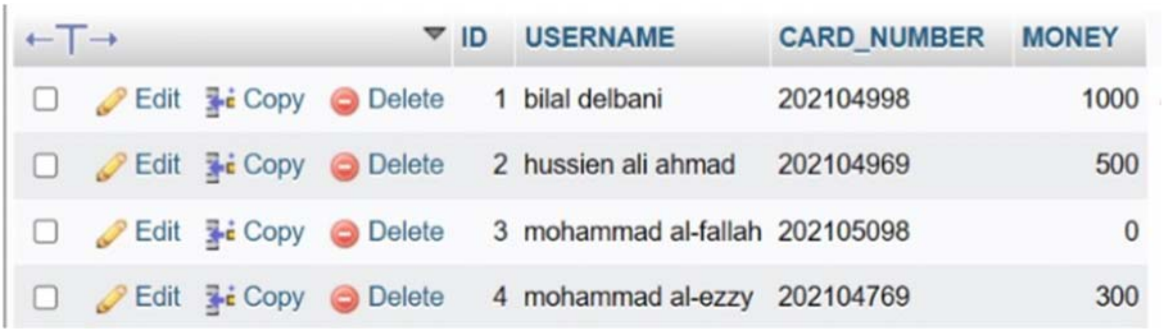


localhost says

Apologies, There is no quantity available right now! Try again in another time...

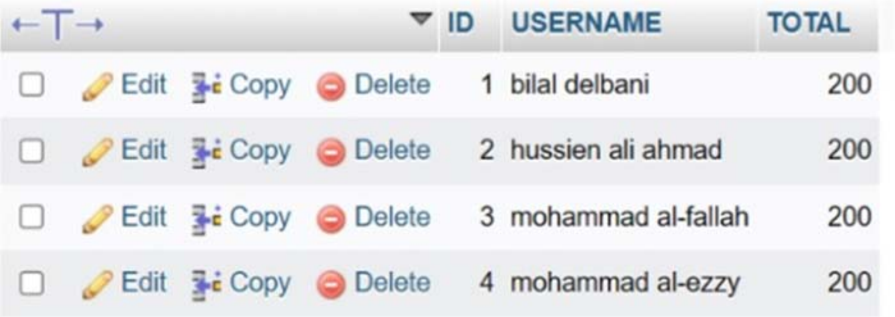
Alert was sent in case the user wants to purchase training bibs because as shown in the above database there are 0 of its quantity.

User Databases:



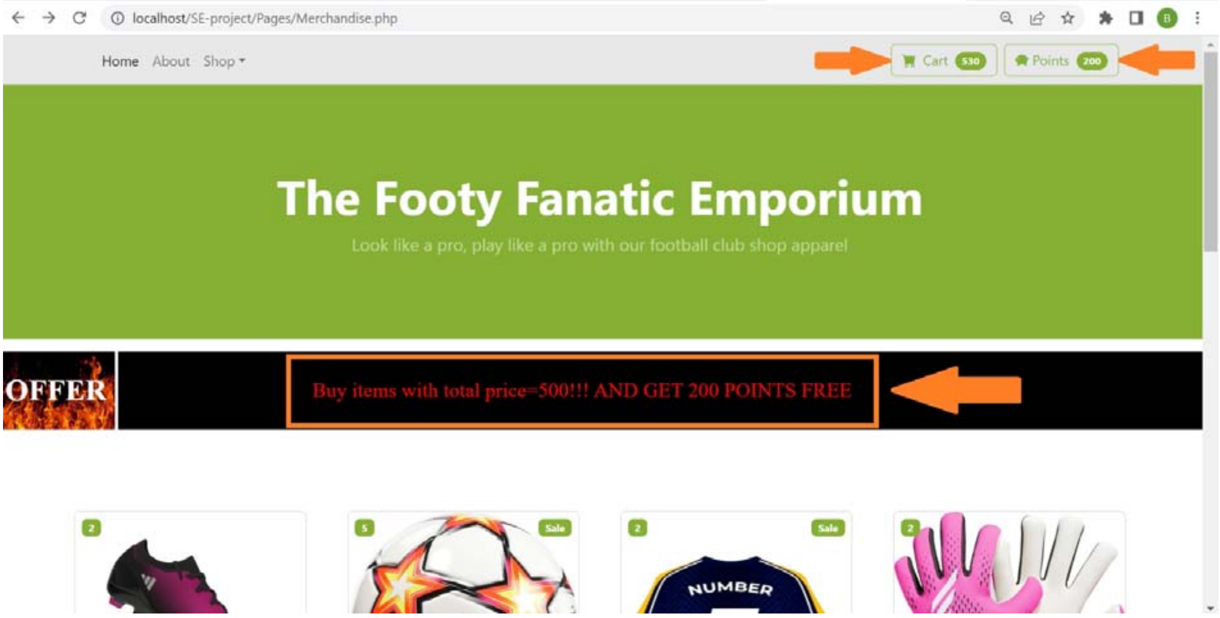
←T→			ID	USERNAME	CARD_NUMBER	MONEY
□	Edit	Copy	Delete	1	bilal delbani	202104998	1000
□	Edit	Copy	Delete	2	hussien ali ahmad	202104969	500
□	Edit	Copy	Delete	3	mohammad al-fallah	202105098	0
□	Edit	Copy	Delete	4	mohammad al-ezzy	202104769	300

Points:



				ID	USERNAME	TOTAL
□	Edit	Copy	Delete	1	bilal delbani	200
□	Edit	Copy	Delete	2	hussien ali ahmad	200
□	Edit	Copy	Delete	3	mohammad al-fallah	200
□	Edit	Copy	Delete	4	mohammad al-ezzy	200

The selected items are worth the following:



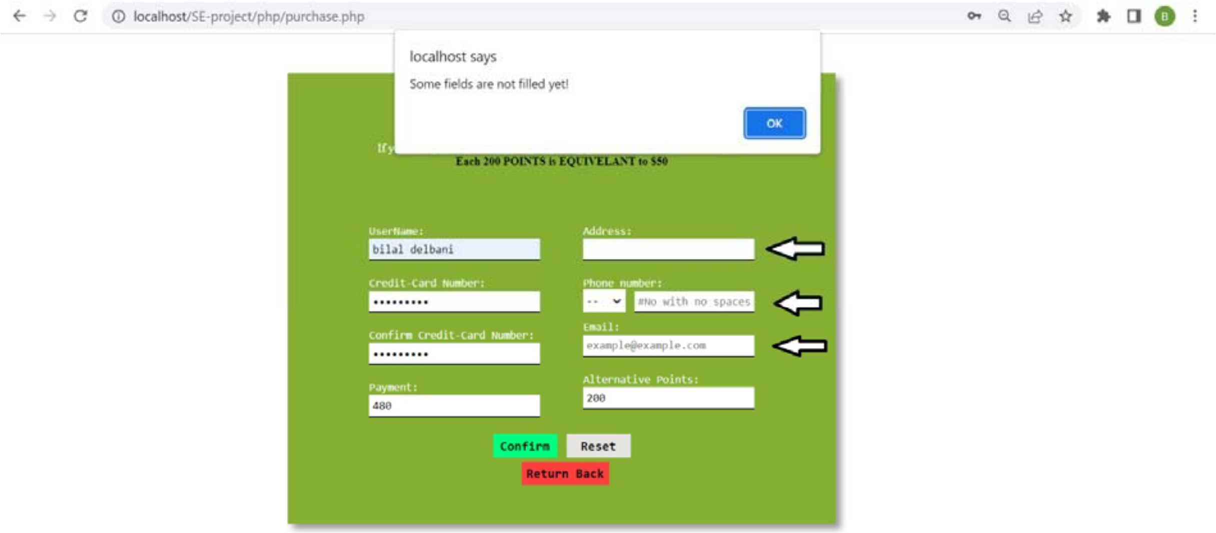
localhost/SE-project/Pages/Merchandise.php

The Footy Fanatic Emporium

Look like a pro, play like a pro with our football club shop apparel

OFFER

Buy items with total price=500!!! AND GET 200 POINTS FREE



localhost/SE-project/php/purchase.php

localhost says

Some fields are not filled yet!

If y

Each 200 POINTS is EQUIVELANT to $50

UserName: bilal delbani

Credit-Card Number: •••••••••

Confirm Credit-Card Number: •••••••••

Payment: 480

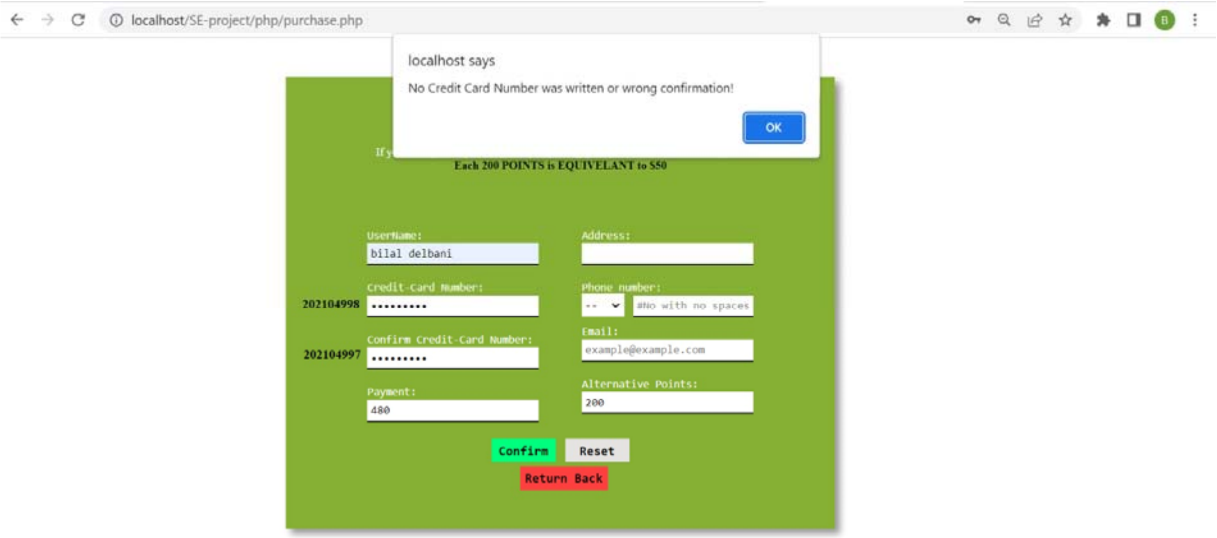
Address:

Phone number: -- #No with no spaces

Email: example@example.com

Alternative Points: 200

In case some fields are missing the user was alerted.


localhost/SE-project/php/purchase.php

localhost says

No Credit Card Number was written or wrong confirmation!

Each 200 POINTS is EQUIVELANT to $50

UserName: bilal delbani

Address:

Credit-Card Number: ••••••••• 202104998

Phone number: -- #No with no spaces

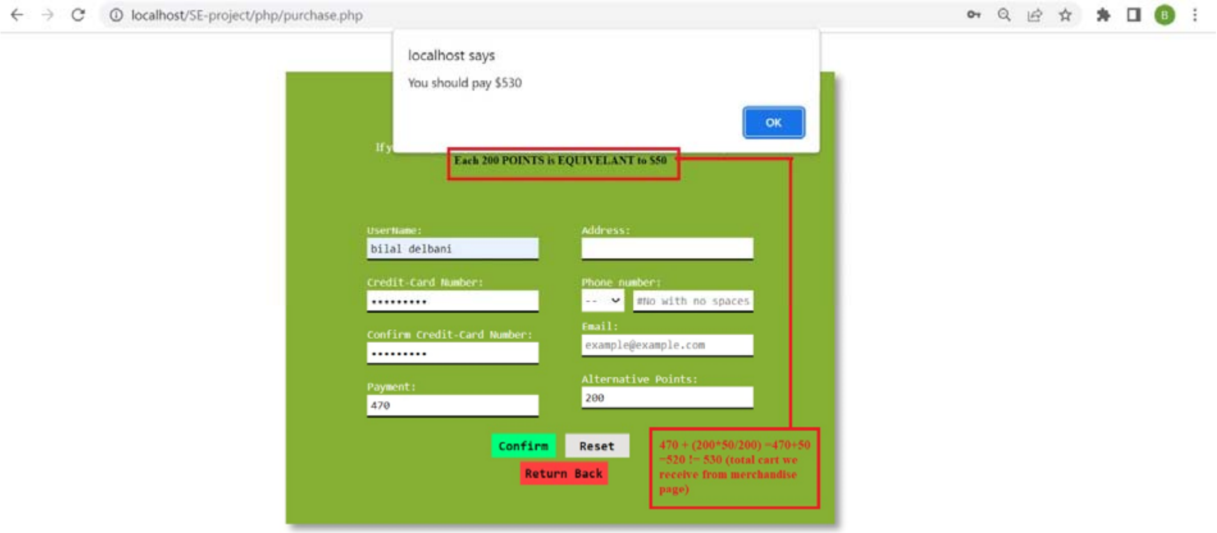
Confirm Credit-Card Number: ••••••••• 202104997

Email: example@example.com

Payment: 480

Alternative Points: 200

In case the credit card info was missing the user was alerted successfully and no transactions were made.


localhost/SE-project/php/purchase.php

localhost says

You should pay $530

Each 200 POINTS is EQUIVELANT to $50

UserName: bilal delbani

Address:

Credit-Card Number: •••••••••

Phone number: -- #No with no spaces

Confirm Credit-Card Number: •••••••••

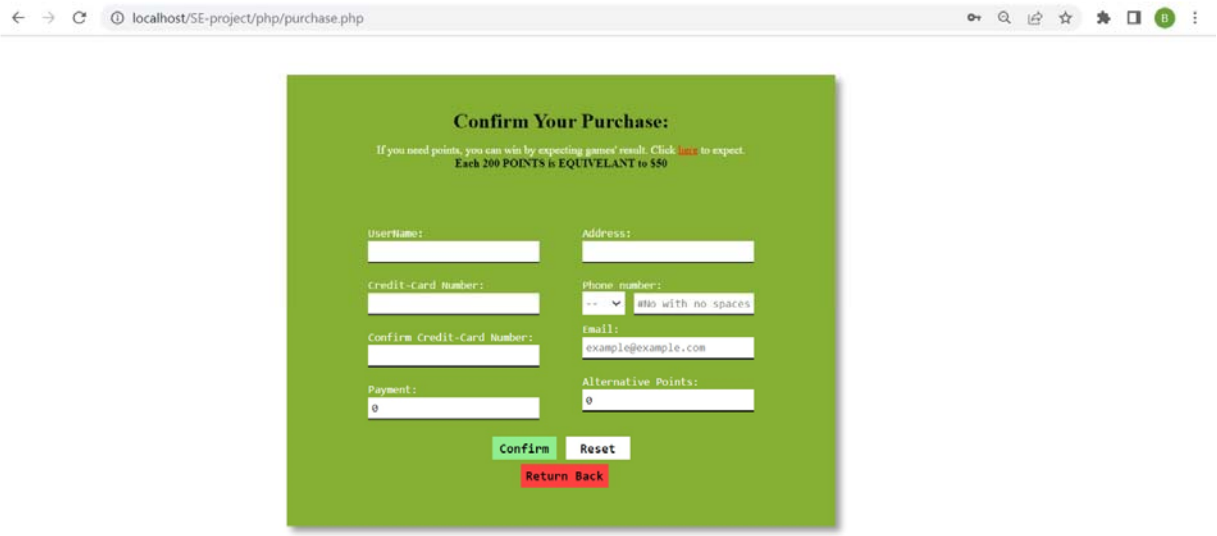
Email: example@example.com

Payment: 470

Alternative Points: 200

470 + (200*50/200) =470+50 =520 != 530 (total cart we receive from merchandise page)

This page shows that when the user pays less than the amount needed he will be alerted because 470 +50 is less than 530.



localhost/SE-project/php/purchase.php

Confirm Your Purchase:

If you need points, you can win by expecting games' result. Click here to expect.
Each 200 POINTS is EQUIVELANT to $50

UserName:

Address:

Credit-Card Number:

Phone number: -- #No with no spaces

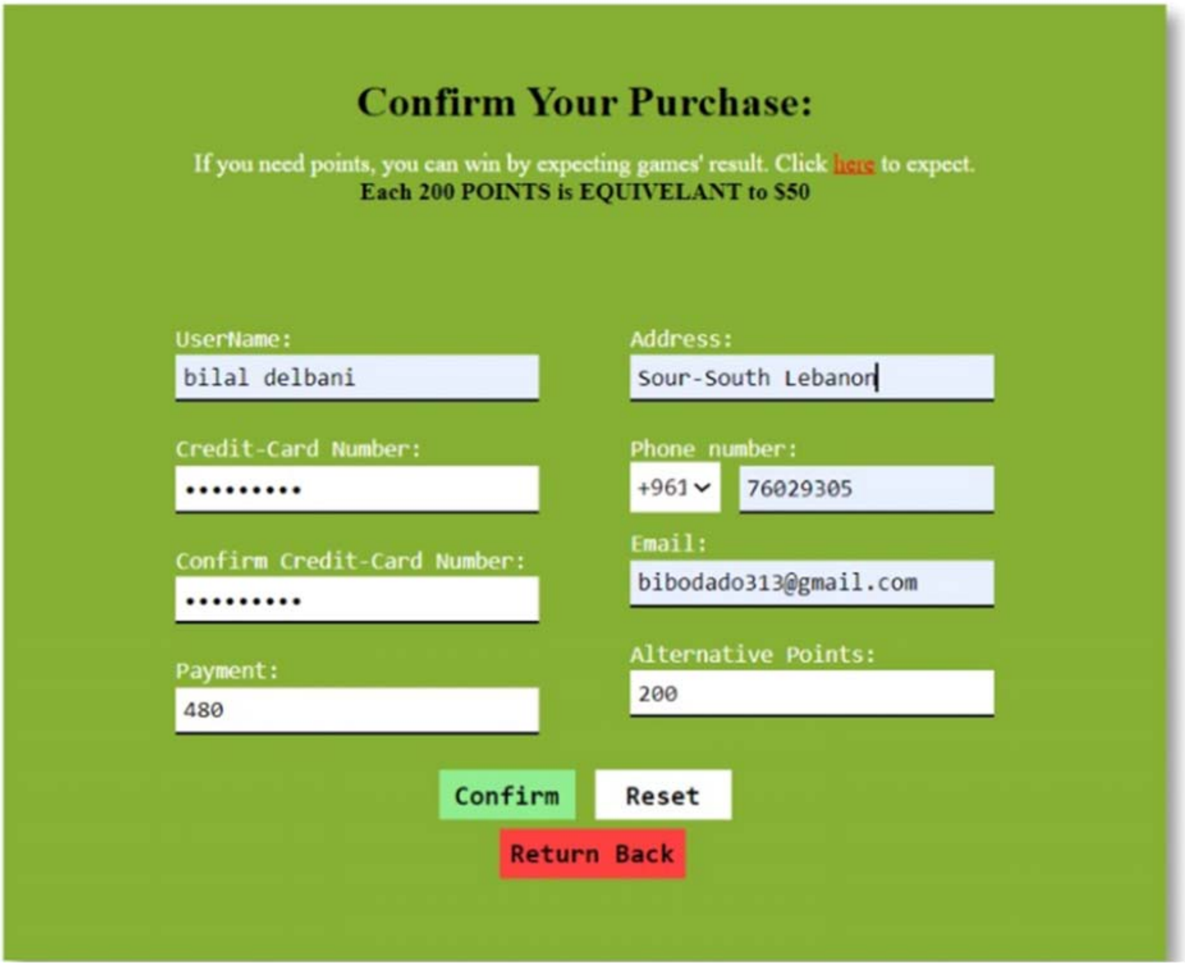
Confirm Credit-Card Number:

Email: example@example.com

Payment: 0

Alternative Points: 0

The reset button works successfully.



Confirm Your Purchase:

If you need points, you can win by expecting games' result. Click here to expect.
Each 200 POINTS is EQUIVELANT to $50

UserName: bilal delbani

Address: Sour-South Lebanon

Credit-Card Number: •••••••••

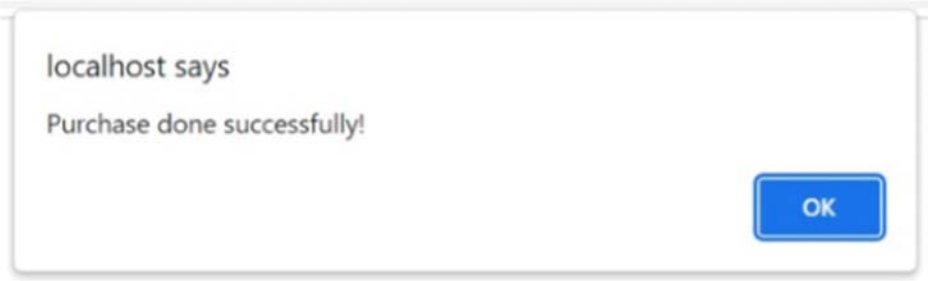
Phone number: +961 76029305

Email: bibodado313@gmail.com

Confirm Credit-Card Number: •••••••••

Alternative Points: 200

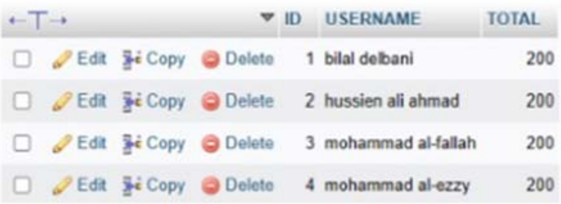
Payment: 480



localhost says

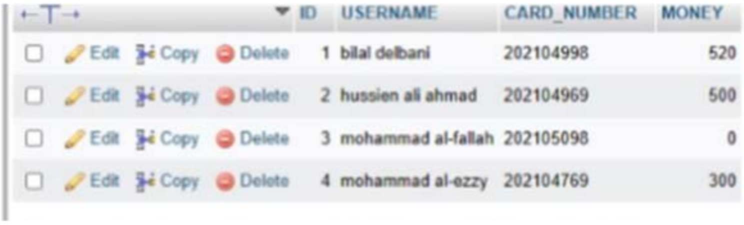
Purchase done successfully!

In case of a successful purchase, the user will be notified. The changes in the database are as follows:



←T→	ID	USERNAME	TOTAL
□ Edit Copy Delete	1	bilal delbani	200
□ Edit Copy Delete	2	hussien ali ahmad	200
□ Edit Copy Delete	3	mohammad al-fallah	200
□ Edit Copy Delete	4	mohammad al-ezzy	200

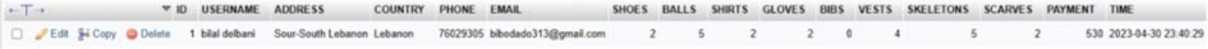
Initially, everyone has 200 points. Since Bilal purchased with 200 points, but he purchased with a price above 500$ his points were recovered.



←T→	ID	USERNAME	CARD_NUMBER	MONEY
□ Edit Copy Delete	1	bilal delbani	202104998	520
□ Edit Copy Delete	2	hussien ali ahmad	202104969	500
□ Edit Copy Delete	3	mohammad al-fallah	202105098	0
□ Edit Copy Delete	4	mohammad al-ezzy	202104769	300

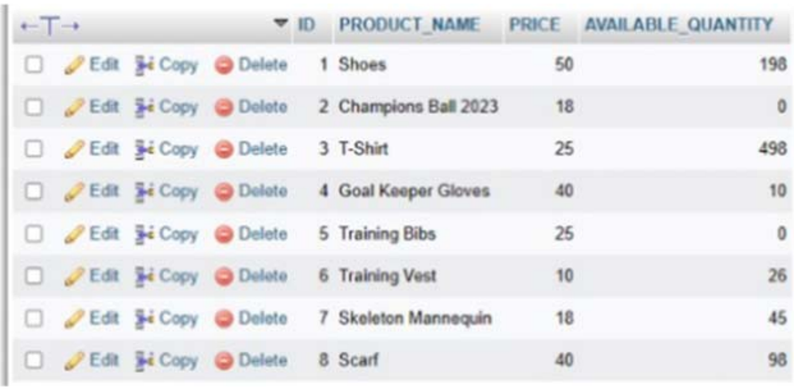
1000-480

Money was removed successfully.



←T→	ID	USERNAME	ADDRESS	COUNTRY	PHONE	EMAIL	SHOES	BALLS	SHIRTS	GLOVES	BIBS	VESTS	SKELETONS	SCARVES	PAYMENT	TIME
□ Edit Copy Delete	1	bilal delbani	Sour-South Lebanon	Lebanon	76029305	bibodado313@gmail.com	2	5	2	2	0	4	5	2	530	2023-04-30 23:40:29

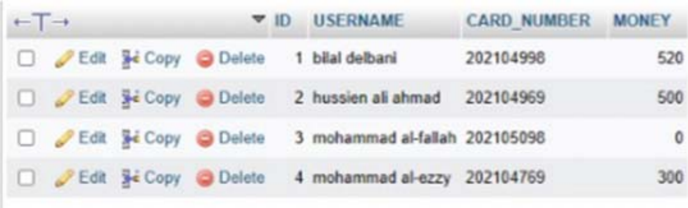
His purchases are shown and added to the database.



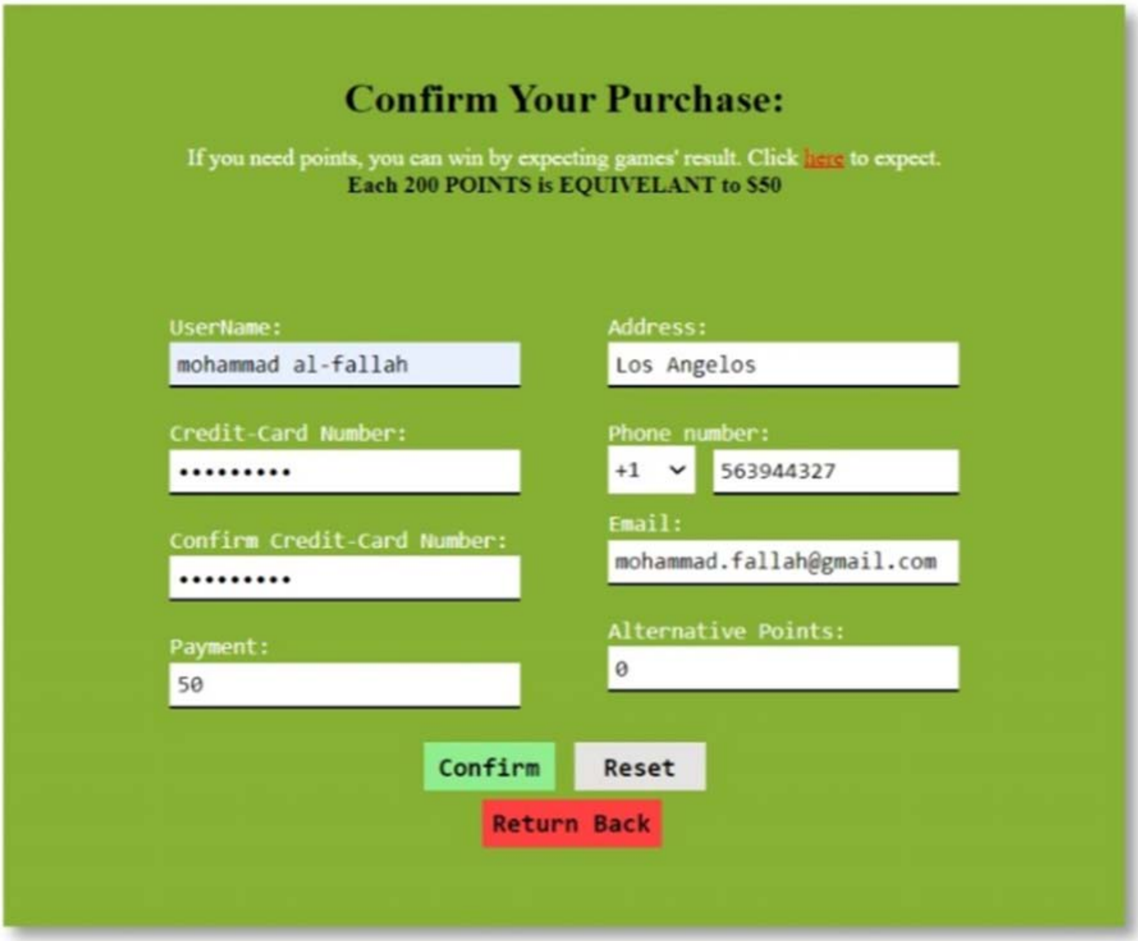
←T→	ID	PRODUCT_NAME	PRICE	AVAILABLE_QUANTITY
□ Edit Copy Delete	1	Shoes	50	198
□ Edit Copy Delete	2	Champions Ball 2023	18	0
□ Edit Copy Delete	3	T-Shirt	25	498
□ Edit Copy Delete	4	Goal Keeper Gloves	40	10
□ Edit Copy Delete	5	Training Bibs	25	0
□ Edit Copy Delete	6	Training Vest	10	26
□ Edit Copy Delete	7	Skeleton Mannequin	18	45
□ Edit Copy Delete	8	Scarf	40	98

The purchased items were removed from the available quantities in the merchandise database.

Testing on a new database:



←T→				ID	USERNAME	CARD_NUMBER	MONEY
□	Edit	Copy	Delete	1	bilal delbani	202104998	520
□	Edit	Copy	Delete	2	hussien ali ahmad	202104969	500
□	Edit	Copy	Delete	3	mohammad al-fallah	202105098	0
□	Edit	Copy	Delete	4	mohammad al-ezzy	202104769	300



Confirm Your Purchase:

If you need points, you can win by expecting games' result. Click here to expect.
Each 200 POINTS is EQUIVELANT to $50

UserName: mohammad al-fallah

Address: Los Angelos

Credit-Card Number: •••••••••

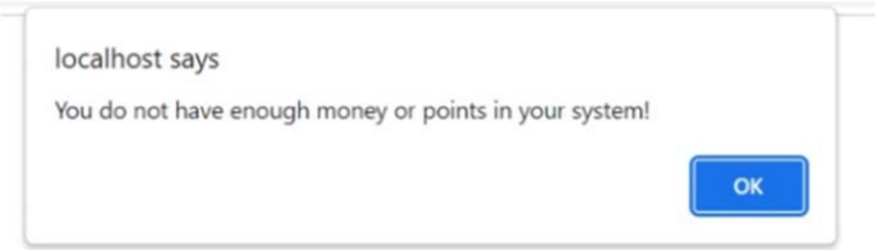
Phone number: +1 563944327

Confirm Credit-Card Number: •••••••••

Email: mohammad.fallah@gmail.com

Payment: 50

Alternative Points: 0



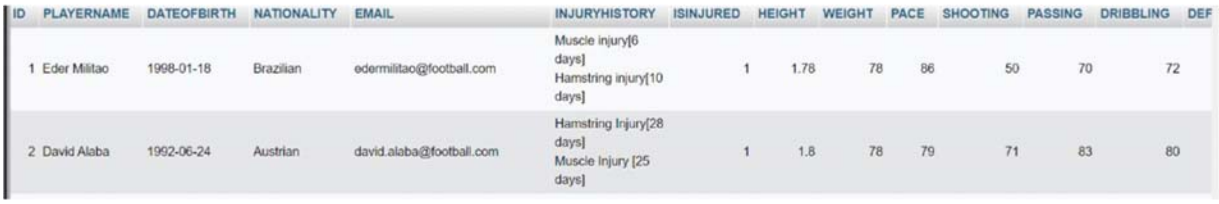
localhost says

You do not have enough money or points in your system!

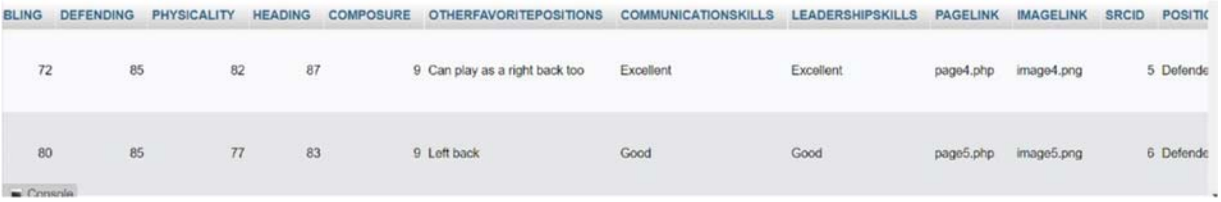
If the user doesn't have enough money, he was alerted successfully and no transactions were made.

Update player’s information:

Database:



ID	PLAYERNAME	DATEOFBIRTH	NATIONALITY	EMAIL	INJURYHISTORY	ISINJURED	HEIGHT	WEIGHT	PACE	SHOOTING	PASSING	DRIBBLING	DEF
1	Eder Militao	1998-01-18	Brazilian	edermilitao@football.com	Muscle injury[6 days] Hamstring injury[10 days]	1	1.78	78	86	50	70	72	
2	David Alaba	1992-06-24	Austrian	david.alaba@football.com	Hamstring Injury[28 days] Muscle Injury [25 days]	1	1.8	78	79	71	83	80	



BLING	DEFENDING	PHYSICALITY	HEADING	COMPOSURE	OTHERFAVORITEPOSITIONS	COMMUNICATIONSKILLS	LEADERSHIPSKILLS	PAGELINK	IMAGELINK	SRCID	POSITIC
72	85	82	87	9	Can play as a right back too	Excellent	Excellent	page4.php	image4.png	5	Defende
80	85	77	83	9	Left back	Good	Good	page5.php	image5.png	6	Defende

1. Invalid player name.

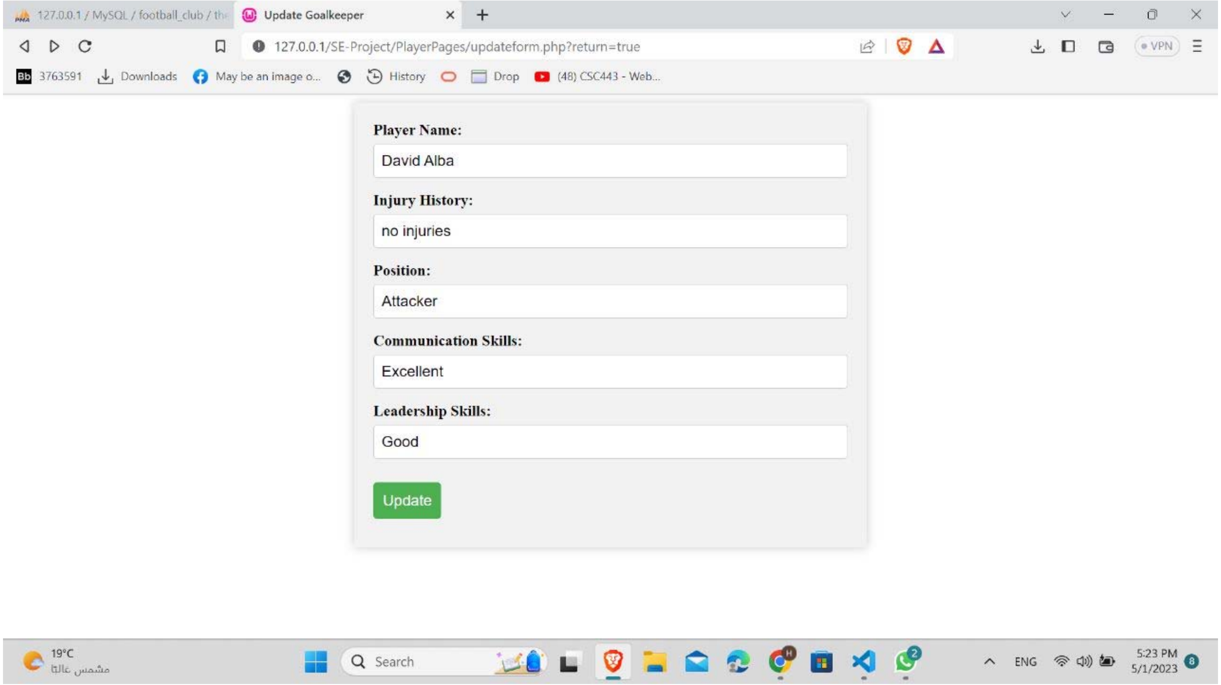
2. Invalid position value.

3. Invalid communication skill value.

4. Invalid leadership value.

5. Successful player’s information update.

1.



127.0.0.1 / MySQL / football_club / th
Update Goalkeeper
127.0.0.1/SE-Project/PlayerPages/updateform.php?return=true
VPN
3763591
Downloads
May be an image o...
History
Drop
(48) CSC443 - Web...

Player Name:

David Alba

Injury History:

no injuries

Position:

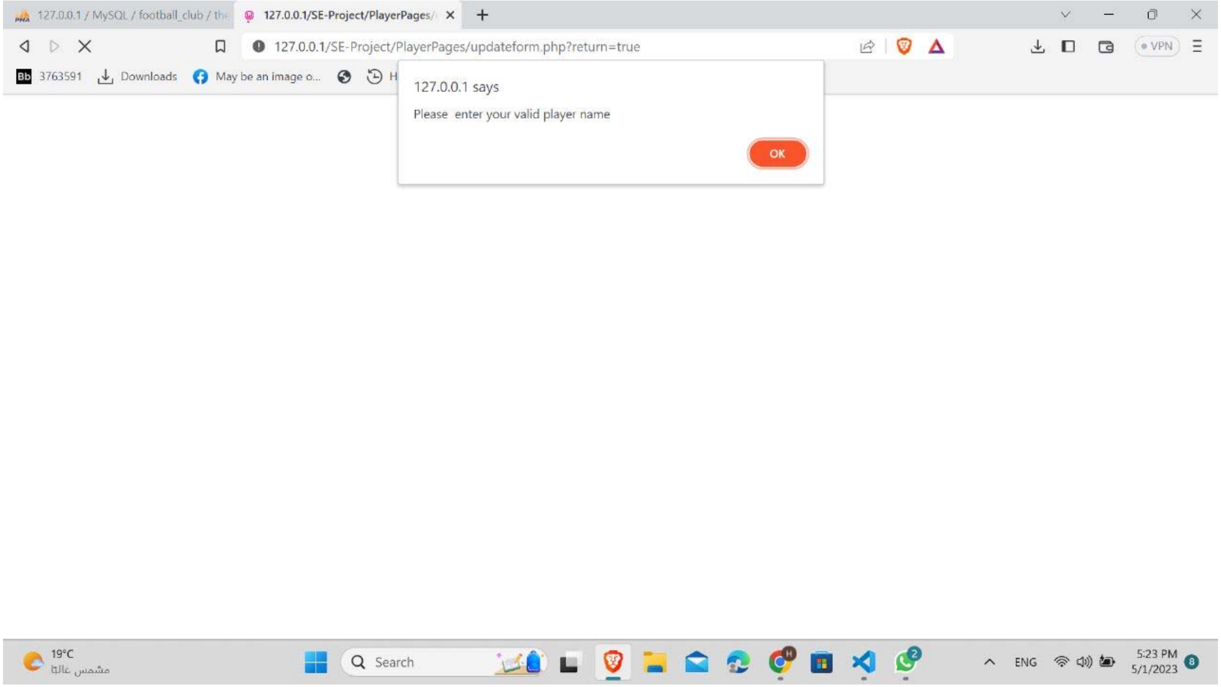
Attacker

Communication Skills:

Excellent

Leadership Skills:

Good

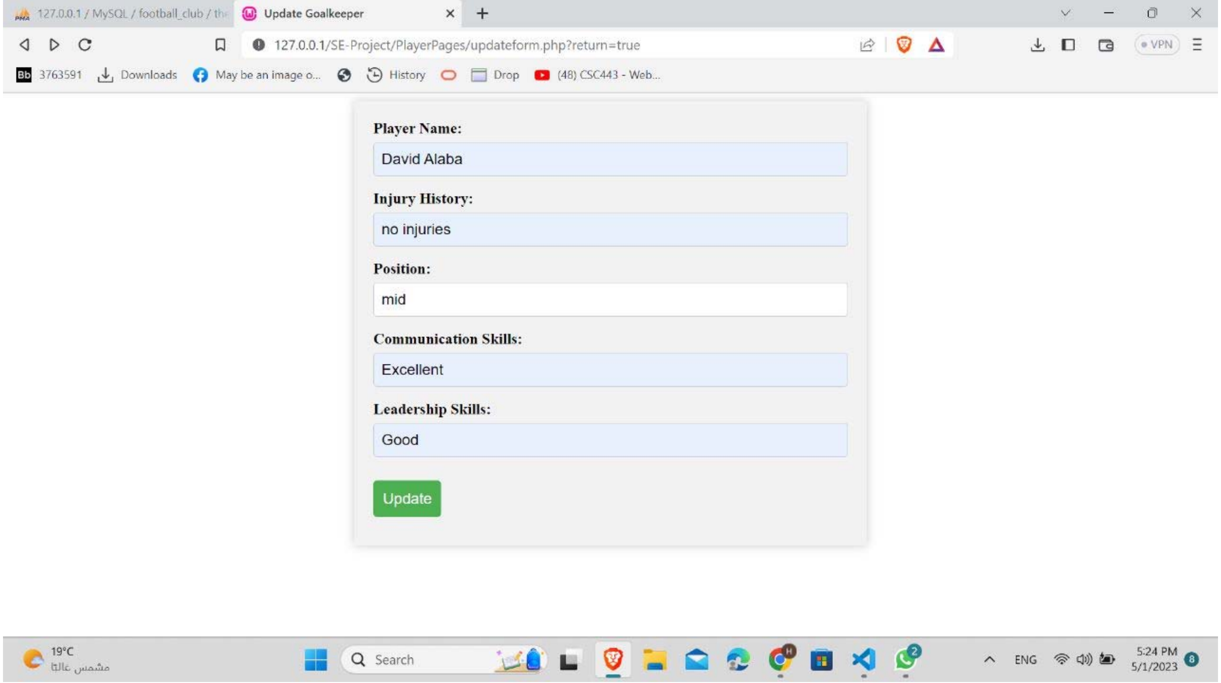

127.0.0.1/SE-Project/PlayerPages/updateform.php?return=true

127.0.0.1 says

Please enter your valid player name

The player name David Alba doesn’t exist, so the manager was alerted successfully.

2.



127.0.0.1/SE-Project/PlayerPages/updateform.php?return=true

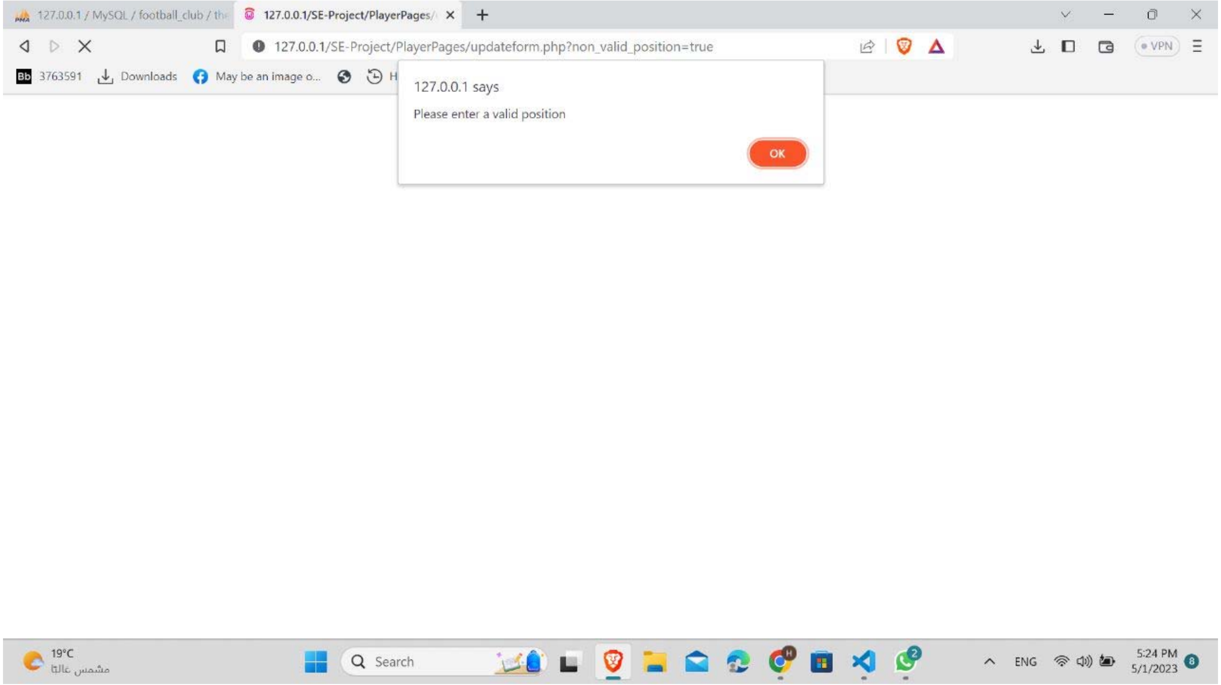
Player Name: David Alaba

Injury History: no injuries

Position: mid

Communication Skills: Excellent

Leadership Skills: Good

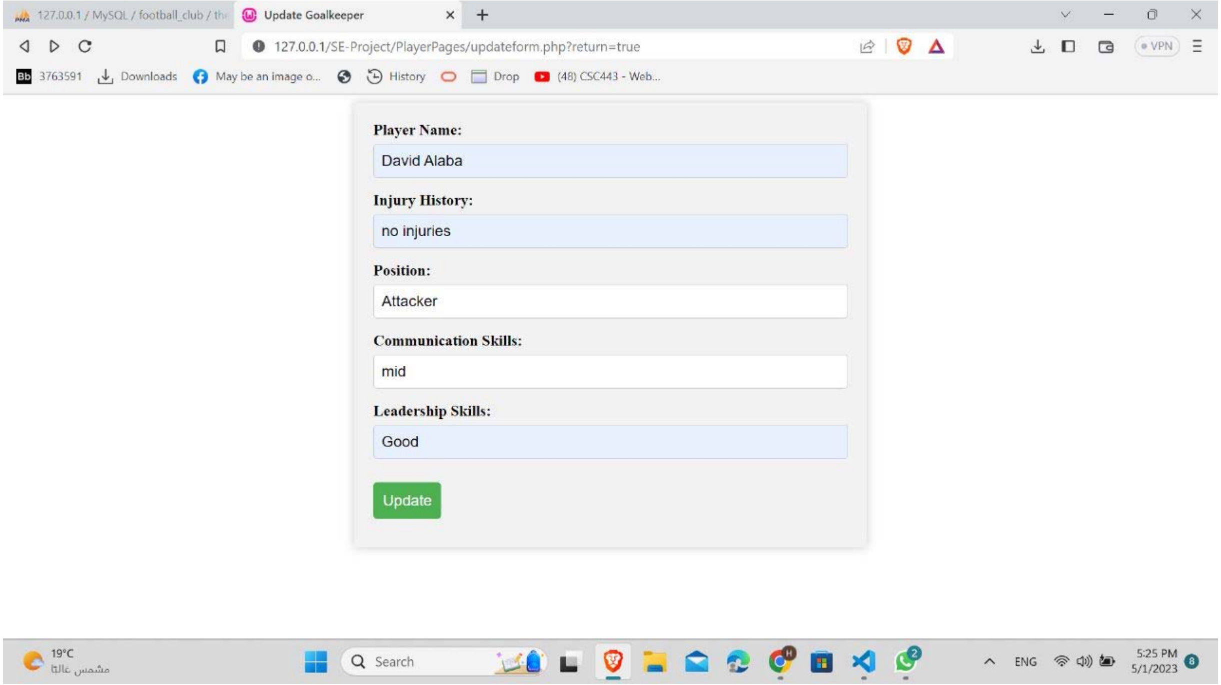

127.0.0.1 / MySQL / football_club / the
127.0.0.1/SE-Project/PlayerPages/
127.0.0.1/SE-Project/PlayerPages/updateform.php?non_valid_position=true
OK
19°C
Search
ENG
5:24 PM
5/1/2023

127.0.0.1 says

Please enter a valid position

The position mid is invalid, and the manager was prevented from updating the database by being alerted successfully.

3.


127.0.0.1 / MySQL / football_club / the
Update Goalkeeper
127.0.0.1/SE-Project/PlayerPages/updateform.php?return=true
Update
19°C
Search
ENG
5:25 PM
5/1/2023

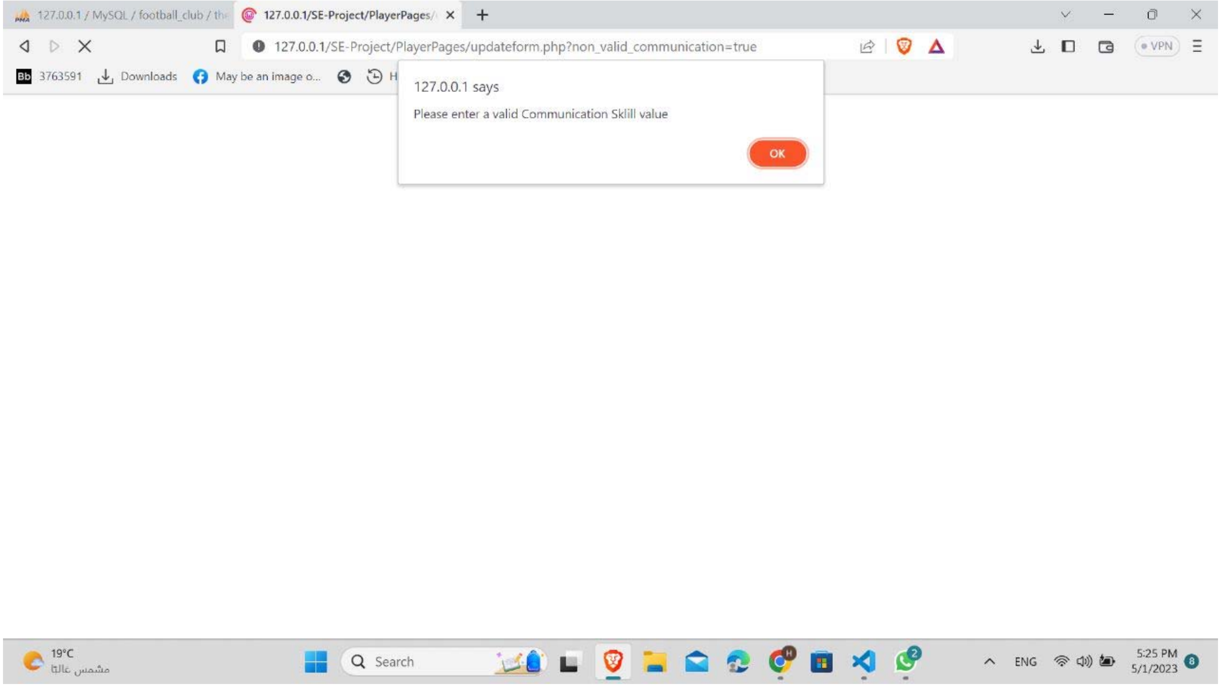
Player Name: David Alaba

Injury History: no injuries

Position: Attacker

Communication Skills: mid

Leadership Skills: Good

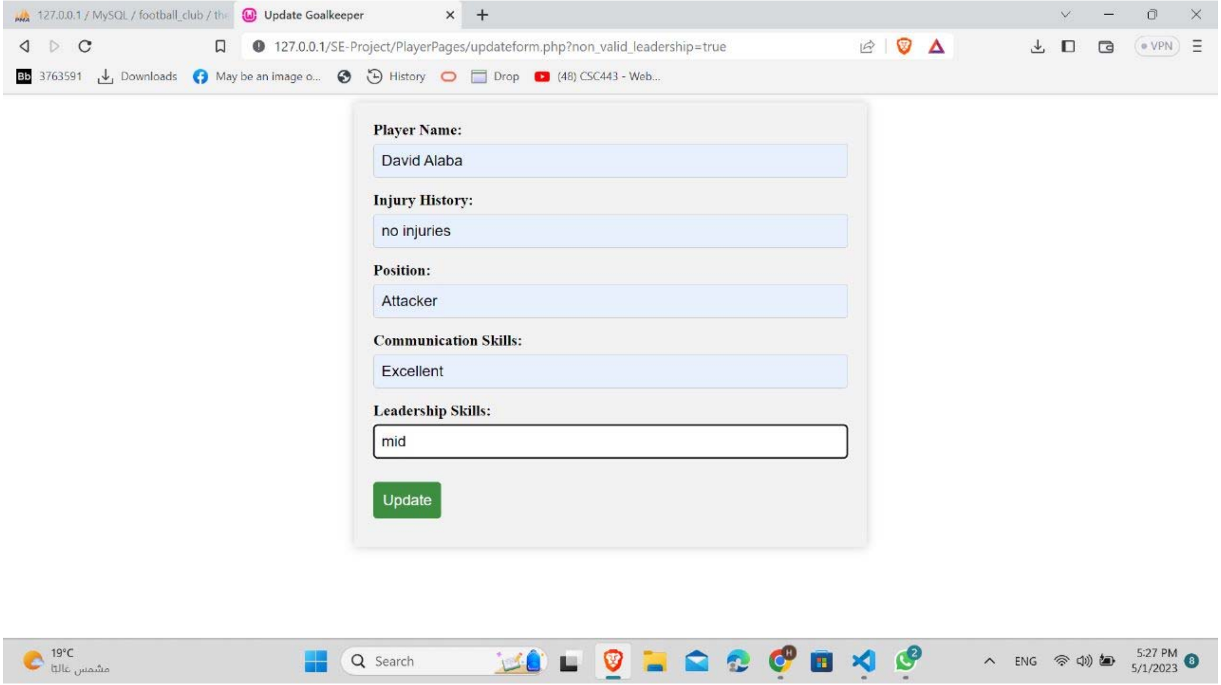

127.0.0.1/SE-Project/PlayerPages/updateform.php?non_valid_communication=true

127.0.0.1 says

Please enter a valid Communication Sklill value

The communication skill value mid is invalid, and the manager was prevented from updating the database by being alerted successfully.

4.


Update Goalkeeper
127.0.0.1/SE-Project/PlayerPages/updateform.php?non_valid_leadership=true

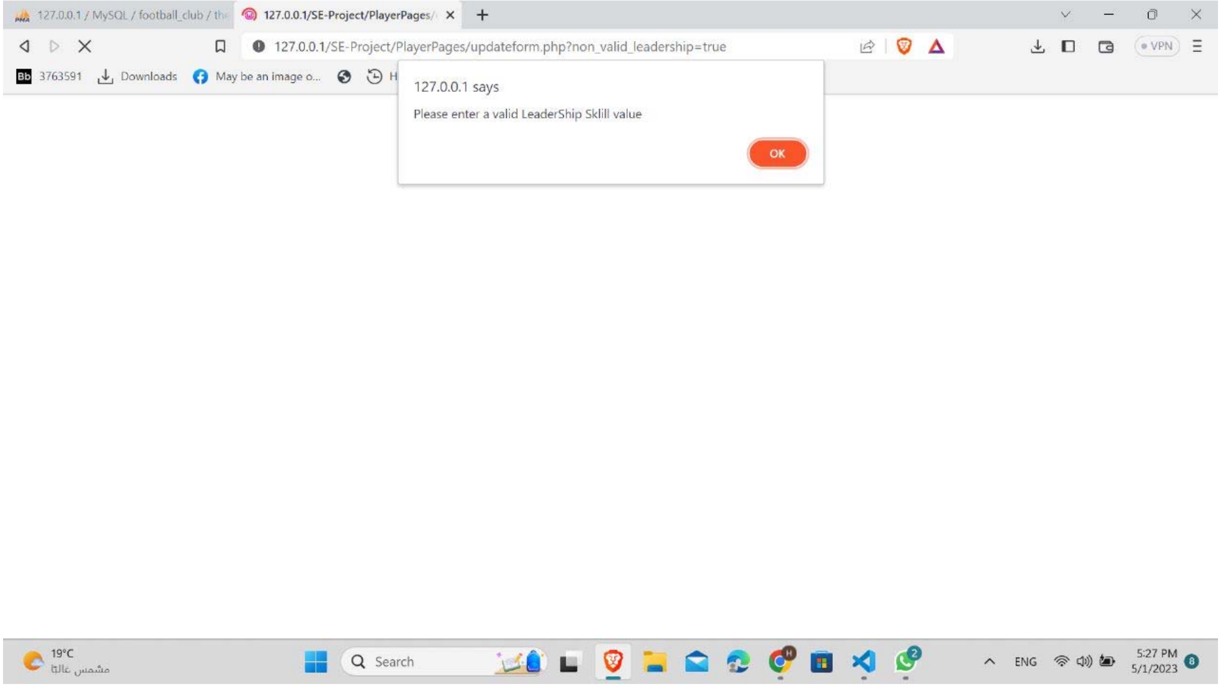
Player Name: David Alaba

Injury History: no injuries

Position: Attacker

Communication Skills: Excellent

Leadership Skills: mid

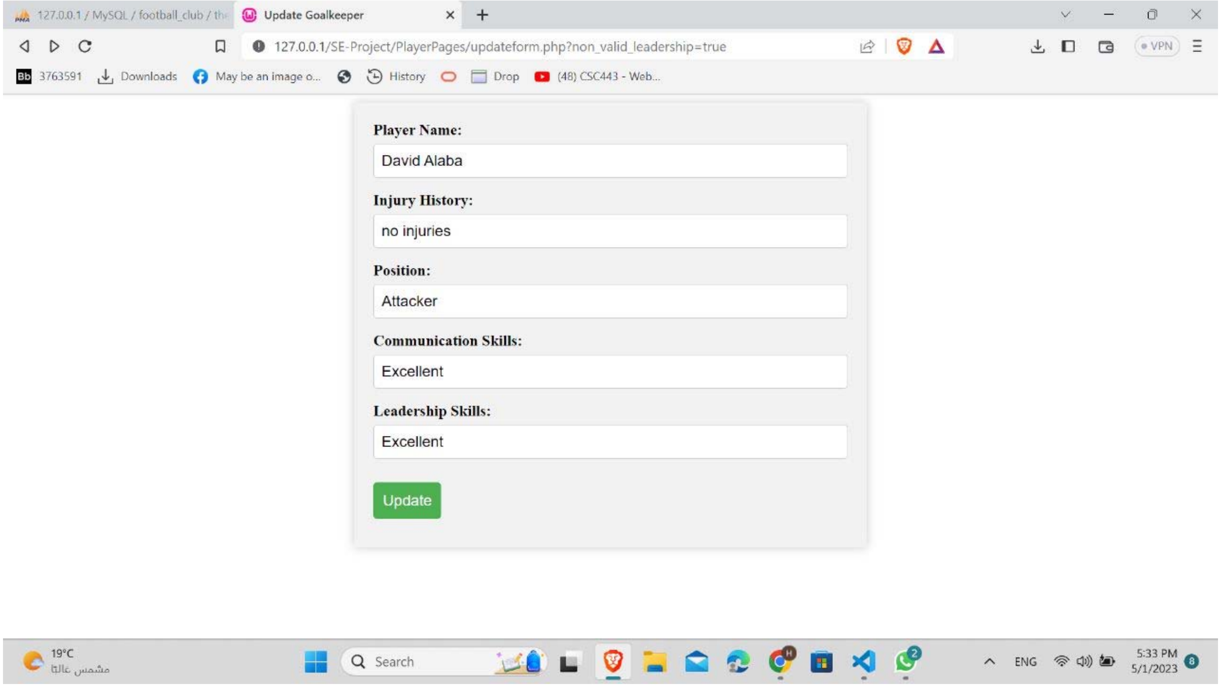


127.0.0.1 says

Please enter a valid LeaderShip Skill value

The leadership skill mid is invalid, so the manager was alerted successfully.

5.



Player Name: David Alaba

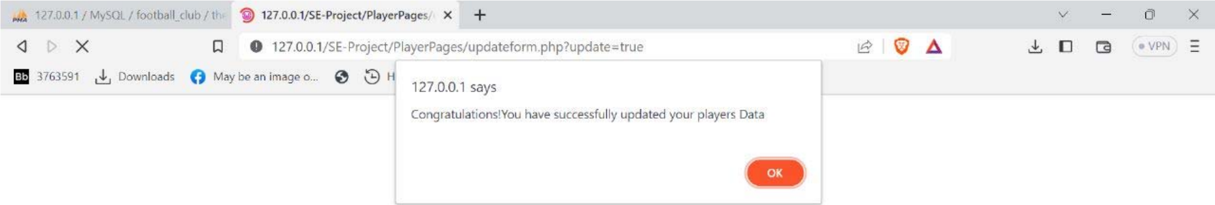
Injury History: no injuries

Position: Attacker

Communication Skills: Excellent

Leadership Skills: Excellent

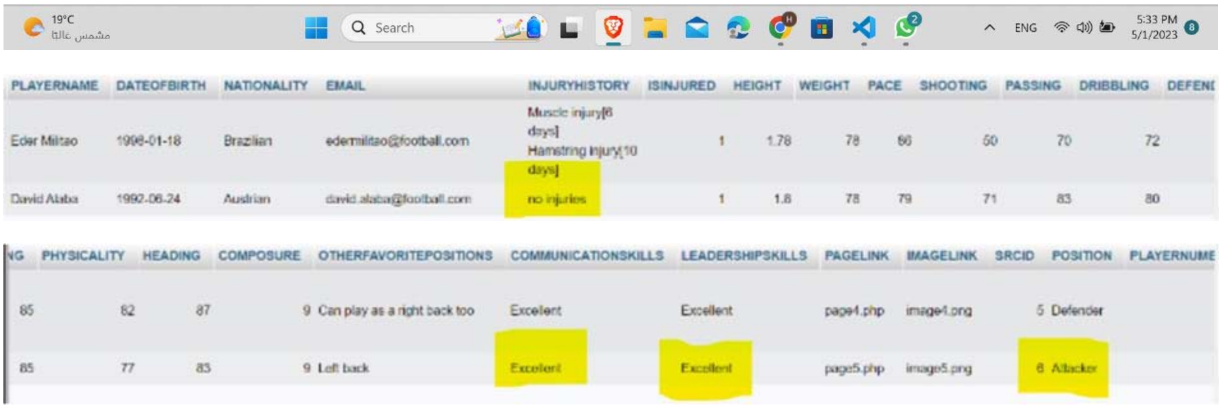
Update Goalkeeper
127.0.0.1/SE-Project/PlayerPages/updateform.php?non_valid_leadership=true
Update


127.0.0.1 / MySQL / football_club / the
127.0.0.1/SE-Project/PlayerPages/
127.0.0.1/SE-Project/PlayerPages/updateform.php?update=true
VPN
3763591
Downloads
May be an image o...
H

127.0.0.1 says

Congratulations!You have successfully updated your players Data

OK



PLAYERNAME	DATEOFBIRTH	NATIONALITY	EMAIL	INJURYHISTORY	ISINJURED	HEIGHT	WEIGHT	PACE	SHOOTING	PASSING	DRIBBLING	DEFENI
Eder Miltao	1998-01-18	Brazilian	edermilitao@football.com	Muscle injury[6 days] Hamstring injury[10 days]	1	1.78	78	66	50	70	72	
David Alaba	1992-06-24	Austrian	david.alaba@football.com	no injuries	1	1.8	78	79	71	83	80	

NG	PHYSICALITY	HEADING	COMPOSURE	OTHERFAVORITEPOSITIONS	COMMUNICATIONSKILLS	LEADERSHIPSKILLS	PAGELINK	IMAGELINK	SRCID	POSITION	PLAYERNUME
85	82	87	9	Can play as a right back too	Excelent	Excellent	page4.php	image4.png	5	Defender	
85	77	83	9	Left back	Excelent	Excellent	page5.php	image5.png	6	Attacker	

When every condition was met successfully, the database was updated accordingly. David Alba recovered from his injury, developed excellent communication and leadership skills, and the manager appointed him as an attacker.

Conclusion:

Finally, the web-based application helps the football club manage a lot of functionalities that are related to its administrators, players, and fans. The implemented functionalities that we provided for this document were signing up, logging in, purchasing merchandise, and updating player information. The use of web technologies and a strong database management system assures that the system is efficient, secure, and user-friendly. Because of its ability to streamline many aspects of a football club's operations, the system is a valuable tool for boosting the club's overall performance.