

# Gym Supplement Website

By:

Abbass Nour Al Deen 105614

Mohammad Abbas Awada 101778

Hussein Al Sharif 105134

Adam Moallem 105821

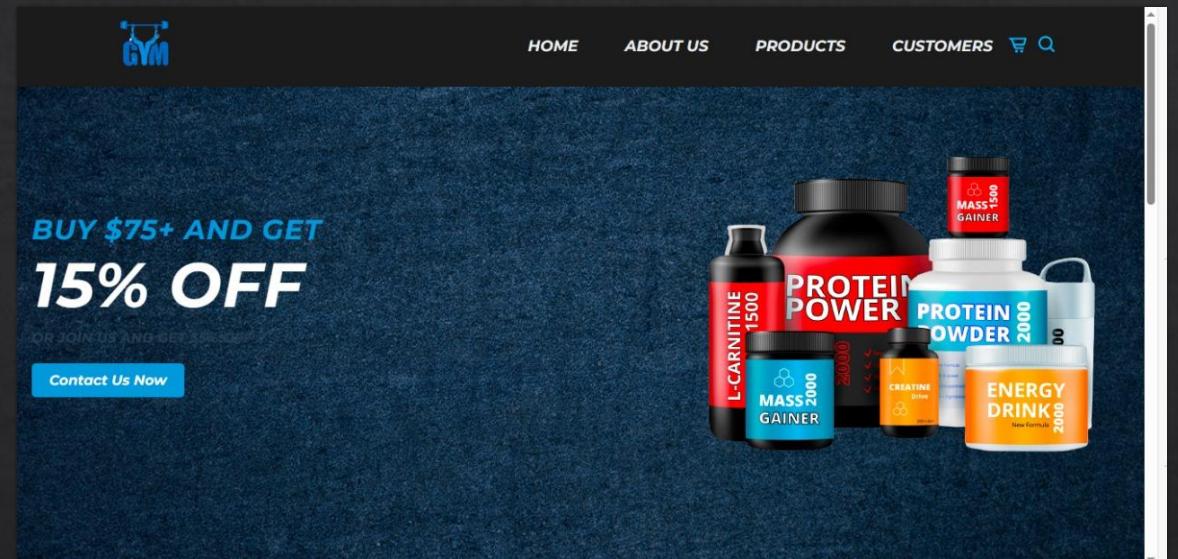
# Introduction

The website's main goal is to provide its customers with gym supplements, while being simple to use and easily navigable.

It is connected to a database that keeps track of the users that have previously made an account using our signup page.

We also used cookies to keep track of items that a user has added to their cart.

Sessions were used to send user information that was submitted via a form to the other pages on the website.



# Login Page

The login page is used to verify whether a user is registered in our database.

Users are required to enter their email and password, and then a query is performed in order to check if the information submitted matches with what's in our database.

The screenshot shows a dark-themed login form for 'GYM SUPPLEMENTS'. At the top, the brand name 'GYM SUPPLEMENTS' is displayed in blue capital letters. Below it is a dark rectangular box containing the login interface. The box has a title 'Login' at the top in light blue. It features two input fields: 'Email' and 'Password', both with placeholder text. Below these fields are two blue buttons: 'Login' on top and 'Sign Up' on the bottom. The entire form is set against a white background, which is part of a larger dark-themed website layout.

GYM SUPPLEMENTS

Login

Email

Password

Login

Sign Up

© 2023 GYM SUPPLEMENTS. All rights reserved.

The user enters their Email and Password, and the query checks if there is a match in our database. And if there's a match then the user logs into their account successfully.

```
if ($t == 0) {
    //add to database
    $add = "INSERT INTO members
(firstName, lastName, phoneNumber, email, password, member, cid)
VALUES
('{$_SESSION["firstName"]}', '{$_SESSION["lastName"]}', '{$_SESSION["phoneNumber"]}',
'{$_SESSION["email"]}', '{$_SESSION["password"]}', 'no', '{$_SESSION["cid"]}')";

    if (!mysqli_query($connection, $add)) {
        echo "error: " . mysqli_error($connection);
    };

    //go to main page
    header("refresh:0;url=index.php");
}
?>
```

If the login information submitted by the user doesn't match anything in the database, then the user is unable to login and is returned to the main page.

```
//wrong password
else {
    echo "<script>
        var toast = document.getElementById('toast');
        toast.innerHTML = 'WRONG PASSWORD';
        toast.style.display = 'block';
        setTimeout(function() {
            toast.style.display = 'none';
        }, 3000);
    </script>";
    break;
}

//if not in database
if ($count == 0) {
    echo "<script>
        var toast = document.getElementById('toast');
        toast.innerHTML = 'WRONG EMAIL';
        toast.style.display = 'block';
        setTimeout(function() {
            toast.style.display = 'none';
        }, 3000);
    </script>";
}

// Free the result set
mysqli_free_result($result);
} else {
    echo "Query failed: " . mysqli_error($connection);
}

?>
```

# Sign Up Page

This page is where the user creates a new account if they do not already have one.

The user enters their first and last names, phone number, email, password, and ID. And then we run a query to make sure the information is valid and isn't already in the database.

After a successful signup the user is redirected back to the main page.

The screenshot shows a sign-up form titled "Sign Up" under a "GYM SUPPLEMENTS" header. The form consists of six input fields: First Name, Last Name, Phone Number, Email, Password, and ID. All fields are currently empty. At the bottom of the form, there is a "Submit" button and a link to "Forgot Password?". The footer of the page contains the copyright notice "© 2023 GYM SUPPLEMENTS. All rights reserved."

The entered information is submitted  
and saved in a session

```
<?php
session_start();

//retrieve data
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $_SESSION["firstName"] = $_POST["firstName"];
    $_SESSION["lastName"] = $_POST["lastName"];
    $_SESSION["phoneNumber"] = $_POST["phoneNumber"];
    $_SESSION["email"] = $_POST["email"];
    $_SESSION["password"] = $_POST["password"];
    $_SESSION["cid"] = $_POST["cid"];
}
```

A query is done to compare the entered information with that in the database.

If Email and ID entered by the user are already in the database, then the user is met with an error message

```
$result = mysqli_query($connection, "SELECT * FROM members");

if ($result) {

    $t = 0;
    //fetching database
    while ($row = mysqli_fetch_assoc($result)) {

        //search if email is in database
        if ($row['email'] == $_SESSION["email"]) {
            $t++;
            echo '<h1>INVALID EMAIL</h1>
                <link rel="stylesheet" href="back.css">
                <button class="btn signup-btn" onclick="goBack()">Go Back</button>
                <script>function goBack() {window.history.back();}</script>';
            // header("refresh:300;url=login.php");
            break;
        }
        if ($row['cid'] == $_SESSION["cid"]) {
            $t++;
            echo '<h1>INVALID ID</h1>
                <link rel="stylesheet" href="back.css">
                <button class="btn signup-btn" onclick="goBack()">Go Back</button>
                <script>function goBack() {window.history.back();}</script>';
            // header("refresh:300;url=login.php");
            break;
        }
    }
}
```

If there are no issues in the signup process, then the user's information is saved in the database, and they are redirected back to the main page.

```
if ($t == 0) {
    //add to database
    $add = "INSERT INTO members
(firstName, lastName, phoneNumber, email, password, member, cid)
VALUES
('$_SESSION["firstName"]'), '$_SESSION["lastName"]', '$_SESSION["phoneNumber"]',
'$_SESSION["email"]', '$_SESSION["password"]', 'no', '$_SESSION["cid"]');
}

if (!mysqli_query($connection, $add)) {
    echo "error: " . mysqli_error($connection);
};

//go to main page
header("refresh:0;url=index.php");
}
?>
```

# Group Page

This page functions as an advertisement for the client to join our group and get 35% OFF on each product.

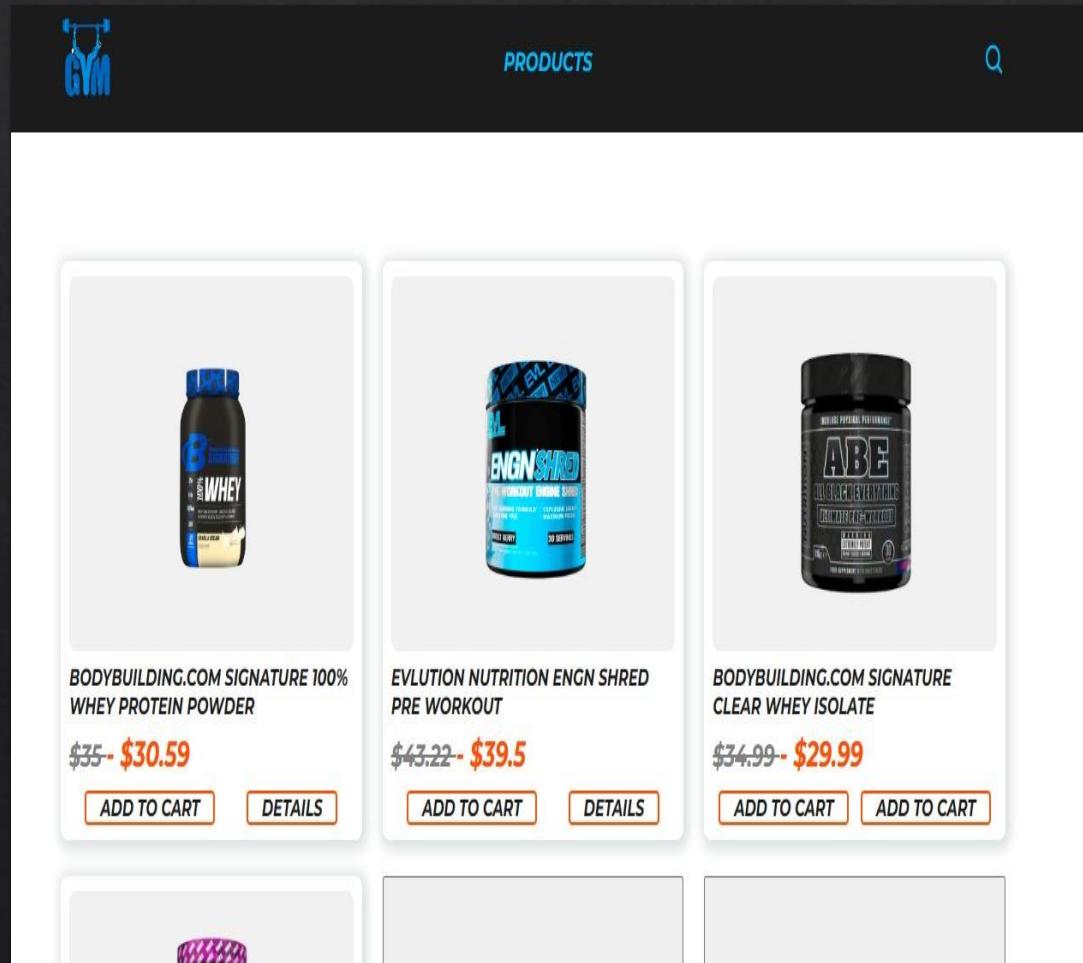
**GYM**

=

**JOIN US FOR ONLY \$25 PER MONTH AND TO GET UP TO 35% OFF FOR ALL PRODUCTS**

**JOIN NOW**   **GO BACK**

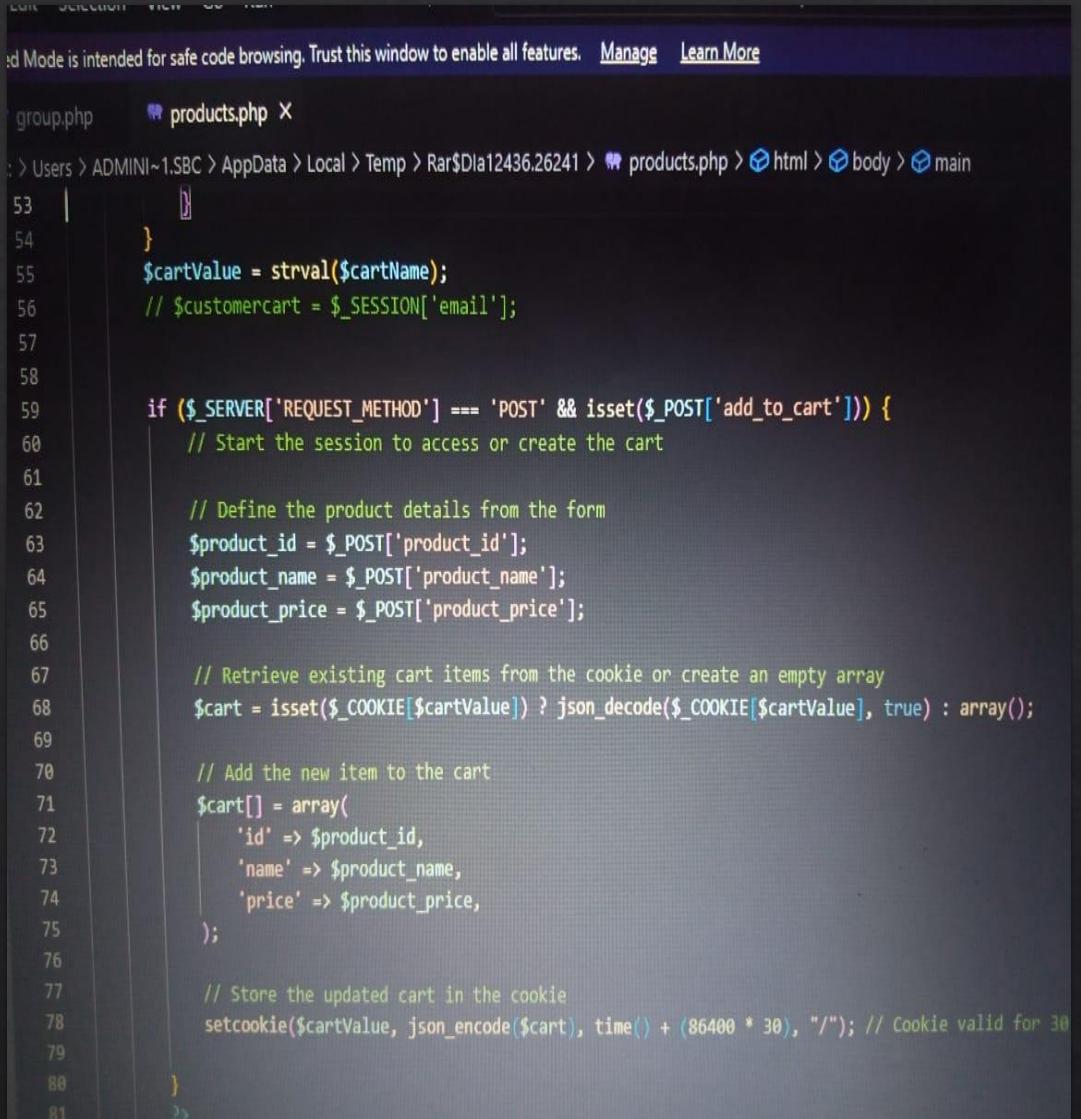
# Products Page



❖ Here comes the products part where the customer can easily scroll through all our products and choose his needs.

❖ For each product, the customer is able to check its price before and after the discount.

- ❖ This piece of code is responsible for adding the products the customer selected into his existing cart, or create a new one if not exist and then add the products.
- ❖ The product(s) he chooses is/are stored in an array each having an id, its name, and price.
- ❖ After that, these information are stored in a cookie for one month ahead from that time
- ❖ Next, we will see an example of a customer's cart.



The screenshot shows a browser developer tools window with the title "products.php". The code is written in PHP and handles the addition of products to a cart via a POST request. It starts by checking if the request method is POST and if a 'add\_to\_cart' key is set. If so, it begins a session and retrieves product details from the POST data. It then retrieves existing cart items from a cookie or creates an empty array. The new product is added to the cart array, which is then encoded as JSON and stored back into a cookie. The cookie is set to expire in 30 days.

```

    <?php
    $cartValue = strval($cartName);
    // $customerCart = $_SESSION['email'];

    if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['add_to_cart'])) {
        // Start the session to access or create the cart

        // Define the product details from the form
        $product_id = $_POST['product_id'];
        $product_name = $_POST['product_name'];
        $product_price = $_POST['product_price'];

        // Retrieve existing cart items from the cookie or create an empty array
        $cart = isset($_COOKIE[$cartValue]) ? json_decode($_COOKIE[$cartValue], true) : array();

        // Add the new item to the cart
        $cart[] = array(
            'id' => $product_id,
            'name' => $product_name,
            'price' => $product_price,
        );

        // Store the updated cart in the cookie
        setcookie($cartValue, json_encode($cart), time() + (86400 * 30), "/");
    }
}

```

# Cart Page

- ❖ In this page, the customer can check what has he selected from our products to buy.
- ❖ He can check the products' names and prices.
- ❖ For example, here our customer chose to buy a whey protein, a preworkout, and an isolate each having its own price.
- ❖ In addition, the client is also able to take a look at the total price before and after the discount.
- ❖ He either proceeds or goes back if any products was added by mistake.

**GYM**

**Shopping Cart**

**Product: whey protien, Price: \$30.59**

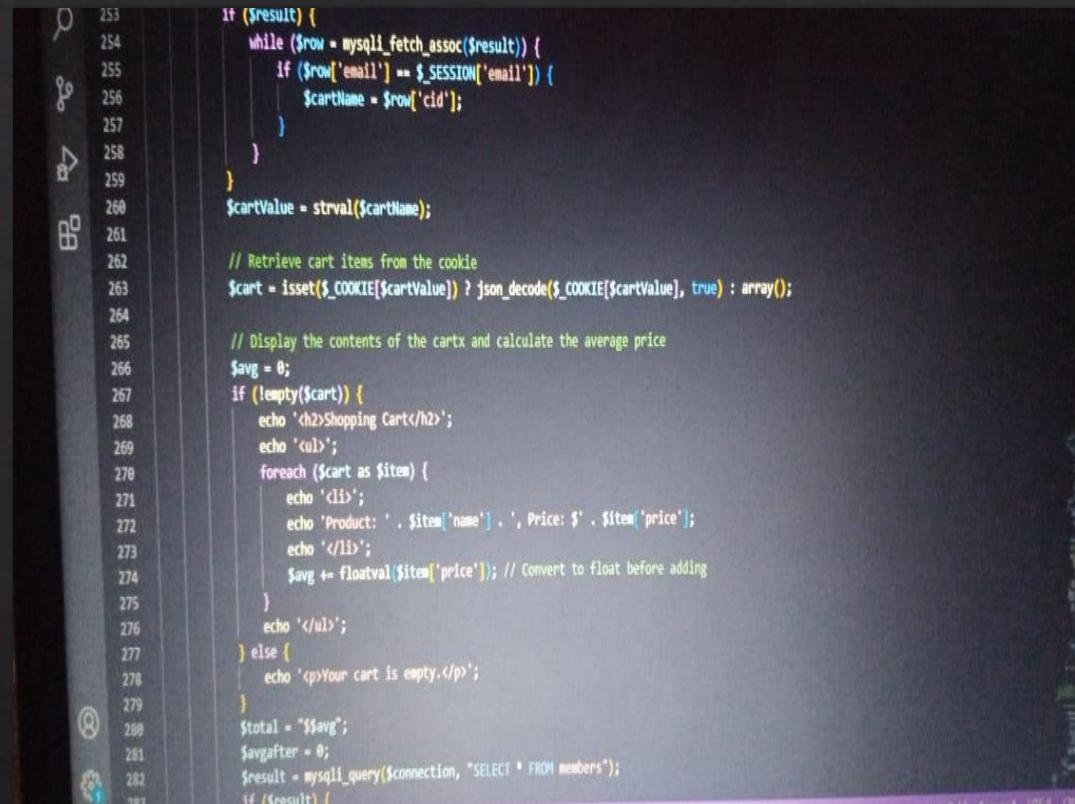
**Product: preworkout , Price: \$39.5**

**Product: isolate , Price: \$\$29.99**

**TOTAL PRICE: \$70.09 - \$24.5315**

**GO BACK**

- ❖ In this piece of code, we first retrieve the customer's products from the cookie we created before.
- ❖ The second step is to loop over them and list each product's name and price.
- ❖ The next step is to sum the total price, and finally calculate the new price after the 35% discount.



A screenshot of a code editor displaying a PHP script. The code is used to handle a shopping cart, starting by retrieving a customer's products from a cookie, then looping through the items to display their names and prices, calculating the average price, and finally summing up the total price. The code includes comments explaining the steps and uses MySQLi functions like `fetch_assoc` and `floatval`.

```
253 if ($result) {
254     while ($row = mysqli_fetch_assoc($result)) {
255         if ($row['email'] == $_SESSION['email']) {
256             $cartName = $row['cid'];
257         }
258     }
259 }
260 $cartValue = strval($cartName);
261
262 // Retrieve cart items from the cookie
263 $cart = isset($_COOKIE[$cartValue]) ? json_decode($_COOKIE[$cartValue], true) : array();
264
265 // Display the contents of the cart and calculate the average price
266 $avg = 0;
267 if (!empty($cart)) {
268     echo '<h2>Shopping Cart</h2>';
269     echo '<ul>';
270     foreach ($cart as $item) {
271         echo '<li>';
272         echo 'Product: ' . $item['name'] . ', Price: $' . $item['price'];
273         echo '</li>';
274         $avg += floatval($item['price']); // Convert to float before adding
275     }
276     echo '</ul>';
277 } else {
278     echo '<p>Your cart is empty.</p>';
279 }
280
281 $total = "$avg";
282 $avgafter = 0;
283 $result = mysqli_query($connection, "SELECT * FROM members");
284 if ($result) {
```

# ViewCart Page

- ❖ Now we are at the end stage of this buying process.
- ❖ The customer is asked to enter his address for our delivery management system.
- ❖ Also, he is free to choose the payment method that best suites him.
- ❖ Last but not least, we concern about our customers' satisfaction. And that's why every customer is free to express his by giving us a feedback.

**Shopping Cart**

*Product: whey protein, Price: \$30.59*  
*Product: preworkout , Price: \$39.5*  
*Product: isolate , Price: \$\$29.99*  
**TOTAL PRICE: \$70.09 - \$24.5315**

**Delivery Information**

**Delivery Address:**

**Payment Method:**

**Submit ORDER**

**GIVE FEEDBACK**

# THE END

- ❖ We as a group, want to show our desire to thank you our grateful Doctor Jaber. We have learnt a lot and a lot in this course specially while working on this project.
- ❖ Thus, we feel it is a must to thank your efforts and wish you the best.

Your beloved students.