

GYM equipment store

Your Cart 0

## Go harder than last time!

Select your gym equipment down below.

Don't forget that if you buy more than 3 items, you get a 10% discount!



**Adjustable Dumbbells**  
EUR100.00

[Add to Cart](#)



**Barbell**  
EUR50.00

[Add to Cart](#)



**Jump Rope**  
EUR10.00

[Add to Cart](#)







So in the header I created a navigation bar that contains two buttons one is the “GYM equipment store” which points back to this main page, the other one is the ”your cart” which I is functioning with javascript with an offcanvas function.

Moreover I created a Hero Section: "Go harder than last time!" with promotional message about 10% discount for 3+ items.

This is like everything is created with bootstrap with a lot of nested </div> elements.

Product Grid: 10 gym equipment items displayed in responsive cards (3 columns on desktop, adapts to mobile) this one I created whit javascript objects which I got from w3schools, It loops through the products array and dynamically generates the HTML for the Bootstrap Cards, injecting them into the product-list container.

For the css part I created a Custom variables (:root) to define consistent colors which I can use later in every part of the page.

For the this main part of the page the Product cards lift up and zoom slightly when hovered.

I created the switchView() function that hides all sections by adding the Bootstrap class d-none. Reveals the target section (Gallery, Checkout, or Confirmation) by removing d-none.

Your Cart	
<b>Barbell</b>	x
EUR50.00	
<b>Adjustable Dumbbells</b>	x
EUR100.00	
<b>Olympic Barbell</b>	x
EUR80.00	
<b>Subtotal:</b>	<b>EUR230.00</b>
<a href="#">Proceed to Checkout</a>	

The `addToCart()` function finds the selected product by ID. Pushes the product object into the cart array.

The `removeFromCart()` function removes the item at the specific index using `cart.splice()`. Updates the UI immediately.

The `updateCartUI ()`function updates the red badge counter in the navbar. Calculates the current subtotal. Regenerates the HTML list inside the Offcanvas. Toggles the "Proceed to Checkout" button state (disabled if the cart is empty).

The `proceedToCheckout()` function closes the Offcanvas menu using the Bootstrap 5. Calls `switchView('view-checkout')` to display the form.

## Shipping Details

Enter your details to complete the order.

FULL NAME	PHONE NUMBER	
<input type="text"/>	<input type="text" value="example: 1234567890"/>	
ADDRESS		
Eötvös utca 60		
CITY	DISTRICT	ZIP
Budapest	<input type="text" value="Choose one ..."/>	<input type="text"/>

[Back to the main page](#) [Complete order](#)

`processOrder()`: Prevent default form submission (stops page reload). `Phone Validation`: Uses Regex to ensure the input contains digits only. `Zip Validation`: Checks if the string length is greater than 4 characters. If everything is okay, it calls `renderConfirmation()`.

`renderConfirmation()`: It Calculates the final sum. `Discount Logic`: Checks if the (`cart.length >= 3`) when it's true it applies a 10% discount and adds a 5% tax and generates a summary list and displays the final total to the user.

For the bootstrap 5 components I used:

**Navbar**: The sticky top header containing the brand and cart trigger.

**Offcanvas**: The hidden sidebar used for the Shopping Cart.

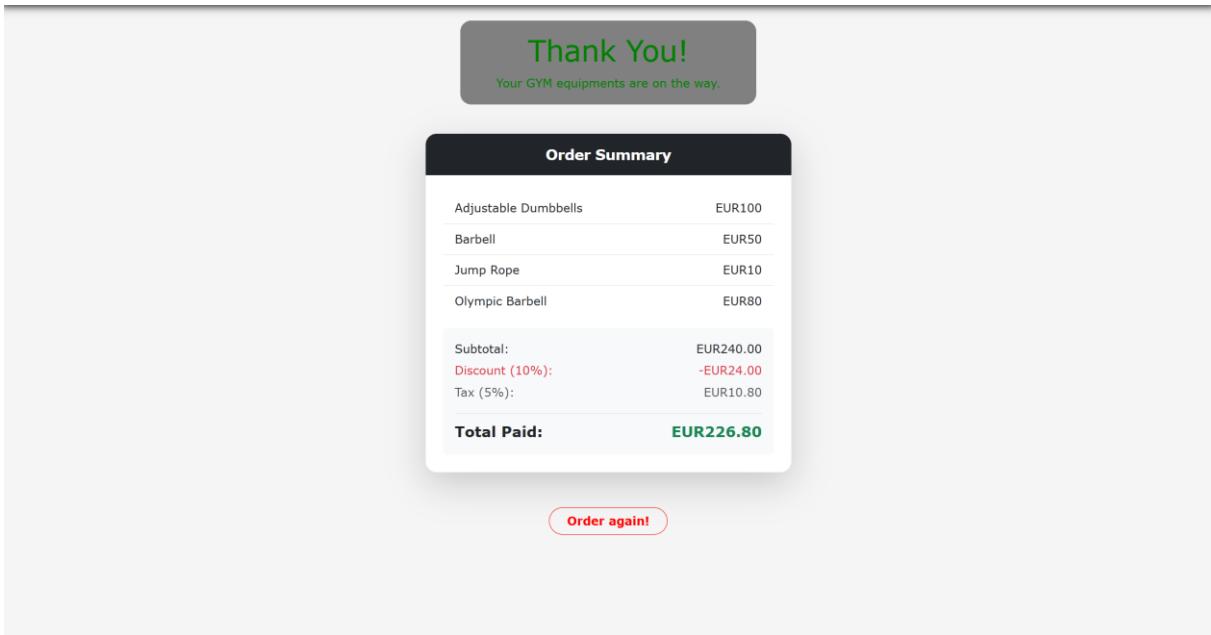
**Cards**: Used to display product details (Image, Title, Price) and the Order Summary.

**Badges**: The red pill-shaped counter on the Cart button.

**Forms (Floating Labels & Validation)**: Used for the checkout process with `is-invalid` feedback classes.

**Alerts**: The success message displayed on the confirmation screen.

**Grid System**: Used throughout the gallery (`col-md-4`) and form layout.



For the final page it displays the order summary and confirms successful purchase with complete pricing breakdown.

The Success Alert displays a large alert with a "Thank You!" message and the Order Summary Card displays a dark header with white text, contains: Each purchased item listed with a product name and a individual price.

For the navigation theres an "Order again!" button that reloads the entire application (location.reload()) but clicking store name in navbar also reloads application.

For the css animations I watched youtube videos like this:

<https://youtu.be/Bhj4miRkSOC?si=mg5TJsw8l13WlCGz>

For the javascript part I used w3schools and I have cousin who is a webdeveloper I asked him for help a few times, but never to write something completely for me.