



Helping Hopeless Programmers

Documentation

Tarik Ayvazi

hhp.tarikajvazi.com

Introduction - General Information

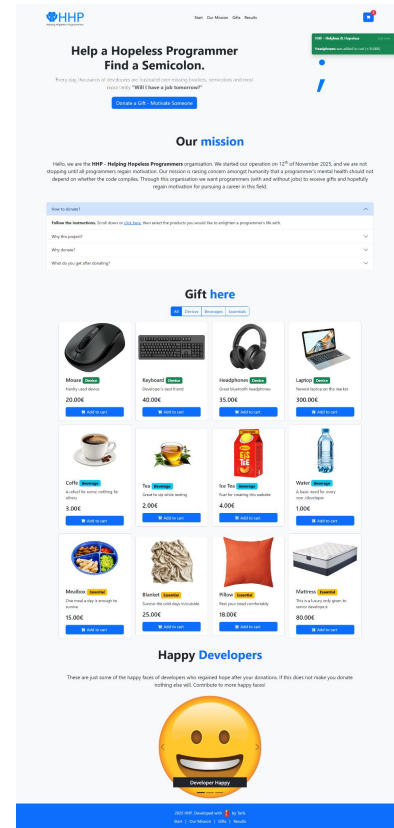
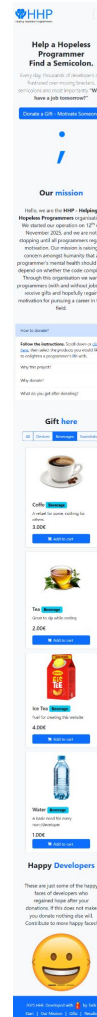
- The idea is about raising money (gifts) for programmers. It is inspired by this unusual 2025 job market where a lot of programmers struggle.
- I used containers and cols for most parts as expected
- I only used CSS for the spinner and the thank you message, so everything else is bootstrap (even the spinner, except its background)
- The logo was custom made
- The first slide of each section will be the workflow, the next slide will be the code

Bootstrap Components

1. Navbar
2. Carousel
3. Accordion
4. Badge (2 uses)
5. Button group as filters
6. Button
7. Cards
8. Close button
9. Collapse for animated checkout
10. Modal
11. Spinner (loader)
12. Toasts (for item added to cart confirmation)
13. Bootstrap forms
14. List group

Landing page

- Simple navbar with the checkout button (icon is svg from bootstrap + badge for current number of items in cart)
- Simple hero section
- Our mission text + Bootstrap Accordion
- Gifts with filters (coded in JavaScript)
- Dynamic display of products from the list of objects
- A carousel of a happy , happier and the happiest programmer as a result of the donations + a footer below

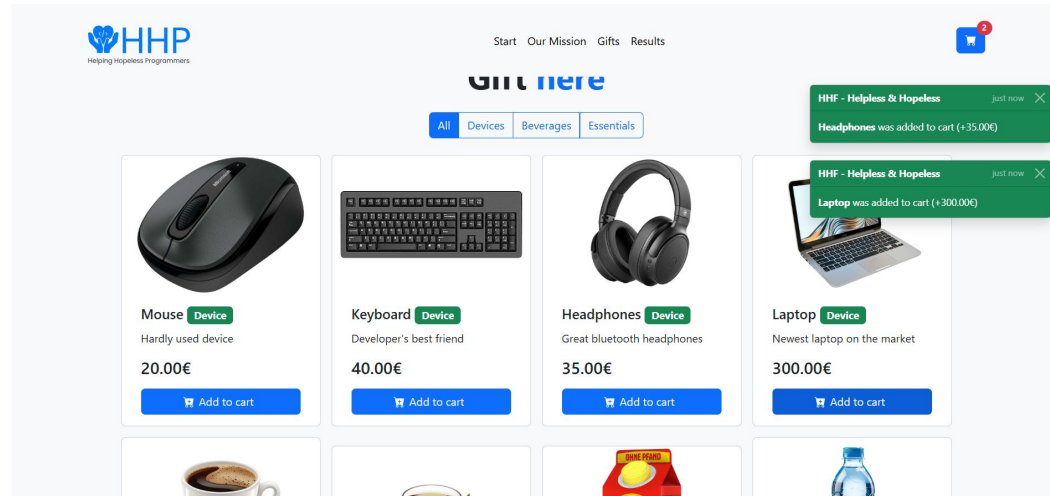


Landing Page - Code Highlights

- For products to displayed I looped through each object in array, and also based on that I assigned some different styling using the switch condition
- For filtering I used the Bootstrap Button Group, and simple javascript functions that hide or show elements with certain classes that were given during the display

Add To Cart

- During the display, each button was given an onclick addToCart(with information parameters). I did not give the id only as I would do in PHP or other languages because it is a small projects and it is easier to retrieve data like this



Add To Cart - Code Highlights




- There is an empty array `cart`, and `addToCart()` adds the object to that cart
- It checks if a product with that name exists, and if yes it only increments the quantity, if not it adds the product
- It then loops through each object in array and sums up their quantities and dynamically displays it in the red circle at the shopping cart
- It also adds a toast notification with a close btn, and information about the name and price

Cart

- When the cart is clicked, this Bootstrap Collapse slides in, and displays in form of list group the **Products** (image name, quantity, price), **Total Details** (price without VAT, VAT, Subtotal, if items \geq 3: discount = 10%, Grand Total).
- Checkout Button** has a function to hide this body canvas and display the personal info

Checkout ×

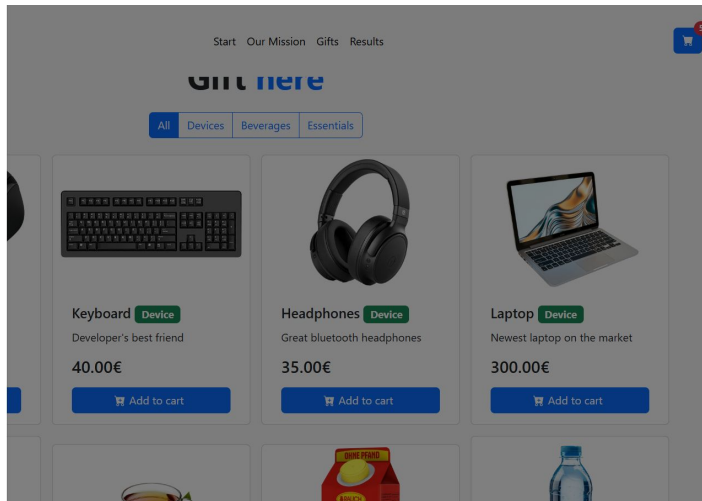
Products

 Headphones	2 x 35.00€
 Laptop	1 x 300.00€
 Keyboard	2 x 40.00€

Total details

Price excl. VAT (-20%)	360.00€
VAT (20%)	90.00€
Subtotal	450.00€
Discount (10%)	45.00€
Grand Total	405.00€

[Checkout](#)

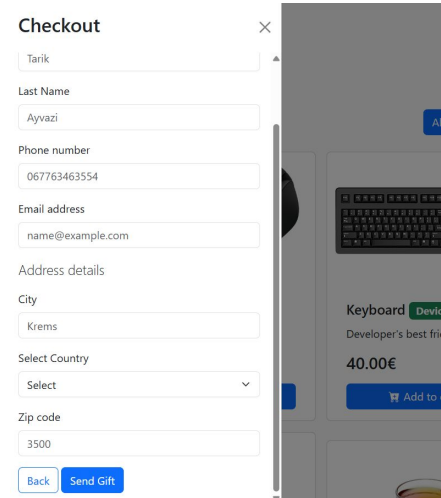


Cart - Code Highlights

- The Total Details and the Checkout Button is added dynamically if `addToCart()` is triggered, because then it would lead to errors (no sense). If there are no products it says No products added.
- Total is calculated by looping through items in `cart[]` and adding to it the `item.quantity * item.product_price`.
- Subtotal is just $0.8 * \text{total}$ (20% Vat)
- VAT is $0.2 * \text{total}$
- Discount is 0 by default, if `item_count` is ≥ 3 then it is 10
- `grandTotal` is calculated including the discount
- Then the list items are just appended to the main ``
- For the numbers i used `.toFixed(2)` for a more professional look

Form

- When Checkout is clicked, we are redirected here. It is required that the fields must be filled with correct information and type of information. If we press back, we go to the cart details, if we press Send Gift, sendGift() is triggered
- sendGift() is meant to store the values of the inputs and open the modal for confirmation, since it cannot have bootstrap and custom function at once, I found parts of code which I tailored to my needs, and made the modal display manually, as well as the validation check



Checkout

First Name

Last Name

Phone number

Email address

Address details

City

Select Country

Zip code

Keyboard

40.00€

Validation

- Bootstrap validations are included.
- The zipcode has maxlength="6" attribute so the input itself does not take more than 6 values.

Checkout



Personal Information

Name

Tarik



Last Name

Ayvazi



Please enter your last name.

Phone number

hello



Please enter a valid phone number.

Email address

tarik



Please enter a valid email address.

Address details

City

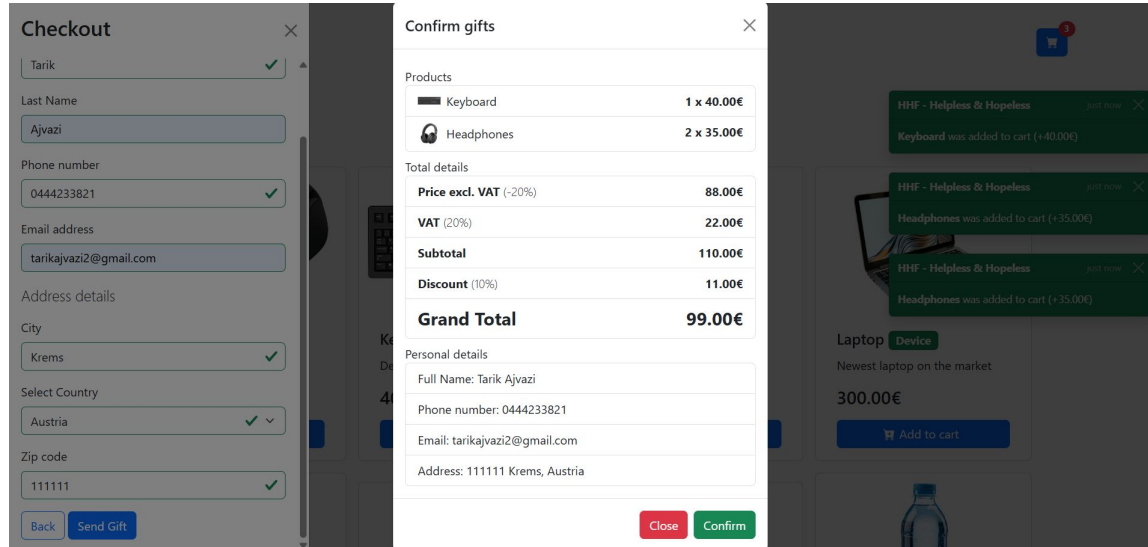
Krems



Select Country

Confirm

- Simply displays all of the information from the Cart details and the forms. Since addToCart() is a different function, it's local variables are not accessible globally, I filled the confirmation box with content when I add something to cart



Checkout

Tarik ✓

Last Name
Ajvazi

Phone number
0444233821 ✓

Email address
tarikajvazi2@gmail.com

Address details

City
Krems ✓

Select Country
Austria ✓

Zip code
11111 ✓

Back Send Gift

Confirm gifts

Products

Keyboard	1 x 40.00€
Headphones	2 x 35.00€

Total details

Price excl. VAT (-20%)	88.00€
VAT (20%)	22.00€
Subtotal	110.00€
Discount (10%)	11.00€
Grand Total	99.00€

Personal details

Full Name: Tarik Ajvazi

Phone number: 0444233821

Email: tarikajvazi2@gmail.com

Address: 111111 Krems, Austria

Close Confirm

HHP - Helpless & Hopeless
Keyboard was added to cart (+40.00€)

HHP - Helpless & Hopeless
Headphones was added to cart (+35.00€)

HHP - Helpless & Hopeless
Headphones was added to cart (+35.00€)

Laptop Device
Newest laptop on the market
300.00€
Add to cart

Thank You

- After Confirm is clicked, it displays a thank you message, with a heart relative to your donation. (I just added 3 conditions in JS and changed the font size).
- There is a button saying See you again, which just redirects you to index.html (current page), but it refreshes the page so that the variables and everything is reset

