

# **Assignment 2 – JavaScript & Bootstrap**

Tatiana Drobca

Department of Science and Technology, IMC University of Applied Sciences Krems

WT1ILV: Web Technologies

Dr. Stefan Klikovits

December 15, 2025



# Veterinary Donation Shop

## Information

### Animal Rescue Veterinary Fund

Provides financial aid for pet owners or rescue groups needing funds for animal medical care, aiming to prevent abandonment by covering costs for treatments, surgeries, or emergency care.

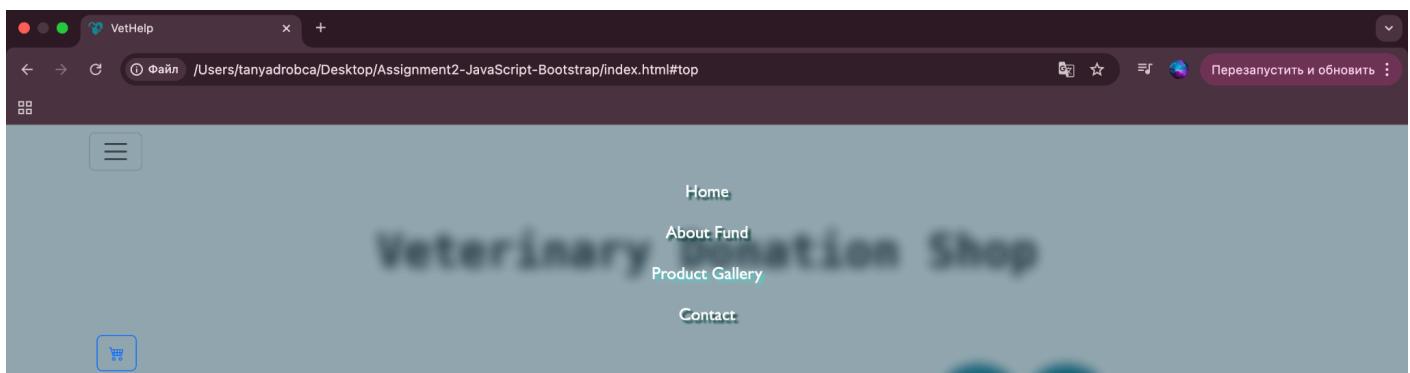


## Product Gallery

Choose service or items to donate - add them to your cart



There are many global problems in our world, and in the search for rational solutions, scientists have compiled the Sustainable Development Goals, but it should be noted that this list does not include items related to human irresponsibility towards animals. On Instagram, I am a member of several volunteer groups and constantly see posts about how people treat their pets, while volunteers work every day to care for abandoned animals and raise money for their treatment and rehabilitation. This problem became the basis for my idea to create a website that provides financial aid for pet owners or rescue groups needing funds for animal medical care, aiming to prevent abandonment by covering costs for treatments, surgeries, or emergency care.



## Information

### Animal Rescue Veterinary Fund

Provides financial aid for pet owners or rescue groups needing funds for animal medical care, aiming to prevent abandonment by covering costs for treatments, surgeries, or emergency care.



For the website, I used Bootstrap to make the header and menu look good and work on all devices. The top navigation uses `<header>` and `<nav>` with classes: `navbar`, `navbar-light`, `sticky-top`, `shadow-sm`. A **burger menu** (`navbar-toggler`) is used, with attributes `data-bs-toggle="collapse"` and `data-bs-target="#burger_nav"`. Clicking it opens the menu using Bootstrap JS: `<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js"></script>`.

The menu links (Home, About Fund, Product Gallery, Contact) are wrapped in `<ul class="navbar-nav ms-auto align-items-lg-center">`. The class `ms-auto` aligns them to the right (but it is not used because on the large screens I use only burger-menue ) Links are styled with `.font_type` - white text, shadow (`text-shadow`), and hover effect ( when shadow color changes).

The cart button uses a Bootstrap icon (`bi bi-cart4`) and shows the number of items dynamically via JS (`<span id="num_p_cart">`).

## Sources:

- Navbar docs: <https://getbootstrap.com/docs/5.3/components/navbar/>
- Bootstrap Icons: <http://icons.getbootstrap.com/>
- Fonts: <https://www.w3.org/Style/Examples/007/fonts.ru.html>

**Product Gallery**

Choose service or items to donate - add them to your cart

<b>Surgical Operation Support</b> Contribution to surgery fund €150.00 <a href="#">Add to Cart</a>	<b>Animal X-Ray</b> Diagnostic radiography €40.00 <a href="#">Add to Cart</a>	<b>Sterilization Support</b> Partial sterilization cost €80.00 <a href="#">Add to Cart</a>	<b>Blood Test</b> Basic blood analysis €28.00 <a href="#">Add to Cart</a>	<b>Ultrasound Examination</b> Ultrasound diagnostics €45.00 <a href="#">Add to Cart</a>
<b>Medication (Antibiotics)</b> Antibiotics course €25.00 <a href="#">Add to Cart</a>	<b>Anti-Parasite Treatment</b> Flea/tick/deworming €30.00 <a href="#">Add to Cart</a>	<b>Vaccination</b> Basic vaccine dose €35.00 <a href="#">Add to Cart</a>	<b>Antiseptics &amp; Bandage Materials</b> Wound care supplies €8.00 <a href="#">Add to Cart</a>	<b>Emergency Vet Visit</b> Urgent consultation €50.00 <a href="#">Add to Cart</a>
<b>Pet Transportation Support</b> Transfer to clinic or shelter €15.00 <a href="#">Add to Cart</a>	<b>Temporary Foster Care Support</b> Daily foster support €25.00 <a href="#">Add to Cart</a>	<b>IV Drip Therapy</b> Infusion therapy session €36.00 <a href="#">Add to Cart</a>	<b>Rehabilitation Therapy</b> Physio/rehab sessions €40.00 <a href="#">Add to Cart</a>	<b>Grooming &amp; Care Supplies</b> Shampoos & brushes €23.00 <a href="#">Add to Cart</a>
<b>Toys For Pets</b> Comfort toys for pets €10.00 <a href="#">Add to Cart</a>	<b>Pet Food Package</b> 1-week food supply €18.00 <a href="#">Add to Cart</a>	<b>Pet Carrier Purchase</b> Carrier for safe transport €35.00 <a href="#">Add to Cart</a>	<b>Warm Blankets for Animals</b> Cozy blankets for shelter €6.00 <a href="#">Add to Cart</a>	<b>Litter Tray Purchase</b> Hygiene tray for cats €12.00 <a href="#">Add to Cart</a>

The product gallery contains **20 different options** that buyers can purchase to support veterinary care. Product cards are dynamically generated using JS from the `products` array. Each card includes: Image, Name, Short description, Price, Add to Cart button (`.add_product()`)JS functions: `add_card()` - renders cards in the gallery, `add_to_cart(by_id)` - adds products to cart



**Anti-Parasite Treatment**

Flea/tick/deworming

€30.00

[Add to Cart](#)

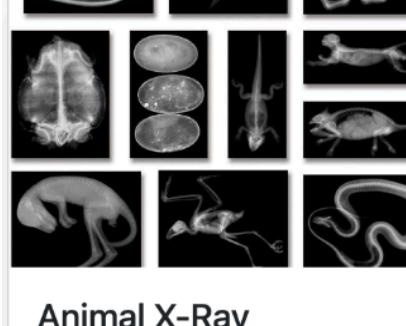


**Surgical Operation Support**

Contribution to surgery fund

€150.00

[Added to cart](#)



**Animal X-Ray**

Diagnostic radiography

€40.00

[Added to cart](#)

When a user clicks Add to Cart, JS updates the cart object and changes the button text to "Added to cart", adding the class `added`. Hover effects are applied: cards `.product-card` lift and show a shadow (`transform: translateY(-6px); box-shadow`).

Grid layout uses Bootstrap `row` and `col-*` with media queries for responsive design: XL: 20% width (`flex: 0 0 20%`), LG: 25% width, XS: 50% width

When each item is added to the cart, the function (`add_to_cart`) is executed.



```

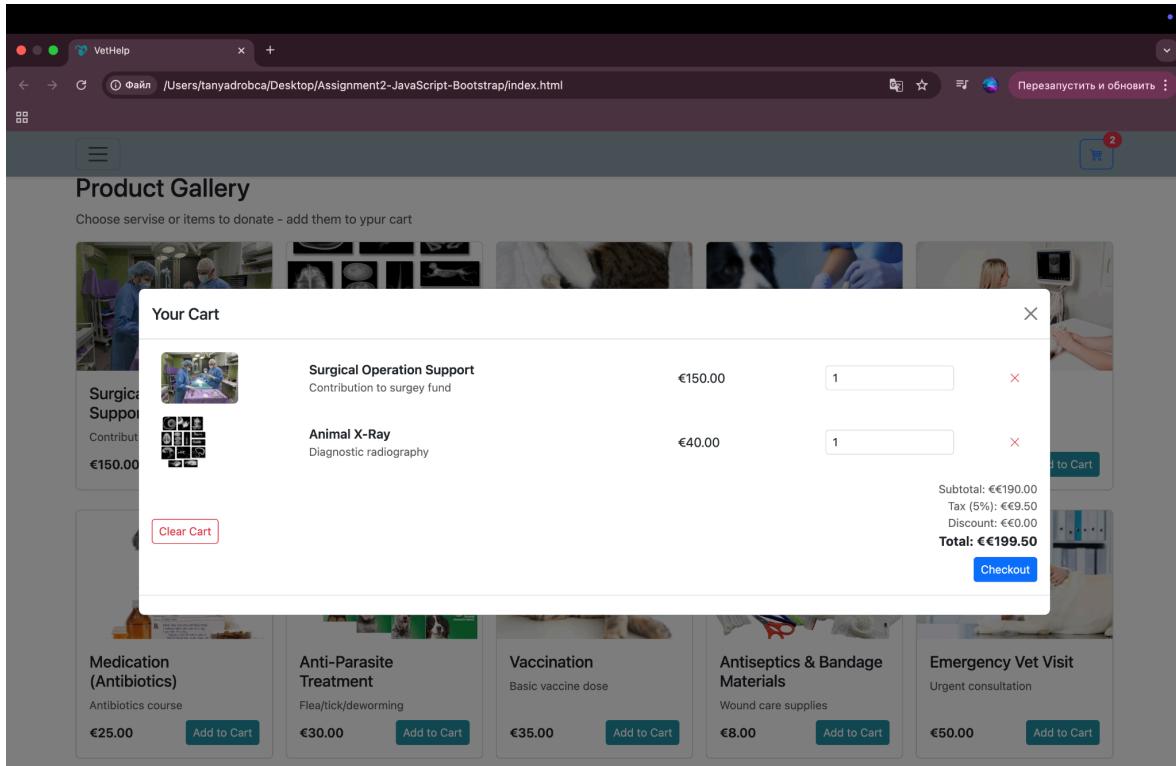
function add_to_cart(by_id){
  if(cart[by_id]) cart[by_id] +=1;
  else cart[by_id] = 1;
  const btn = document.querySelector(`.add_product[data-id='${by_id}']`);
  if(btn){
    btn.classList.add('added');
    btn.textContent = 'Added to cart';
  }
  update_cart();
}

function update_cart(){
  const count = Object.values(cart).reduce((s,q)=>s+q,0);
  num_p_cart.textContent = count; // the num of products displays on the screen in icon
  num_p_cart.style.display = count>0 ? 'inline-block': 'none'; // if there is nothing displays 0
}

```

## Sources:

- Tutorial on dynamic card creation: [https://youtu.be/z\\_mTYpx49v4](https://youtu.be/z_mTYpx49v4)
- One more tutorial: <https://youtu.be/gXWohFYrI0M?si=Bznxt6IkK5EvOos6>



```
<div class="modal fade" id="shopping_cart" tabindex="-1" aria-hidden="true"> <!--https://getbootstrap.com/docs/4.0/components/modal/>
    <div class="modal-dialog modal-xl modal-dialog-centered">
        <div class="modal-content">
            <div class="modal-header">
                <h5 id="cart_head" class="modal-title">Your Cart</h5>
                <button class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div id="cart_body" class="modal-body"></div>
            <div id="cart_footer" class="modal-footer"></div>
        </div>
    </div>
</div>
```

The shopping cart is the most complex part of the project, as it involves the largest number of JS elements. Implemented as a Bootstrap modal (`modal`, `fade`, `modal-xl`, `modal-dialog-centered`).

```
function add_card(){
    product_gallery.innerHTML = '';
    products.forEach(p => {
        const col = document.createElement('div');
        col.className = 'col-6 col-sm-6 col-md-4 col-lg-3 card-col d-flex';
        col.innerHTML = `
            <div class="card product-card w-100">
                
                <div class="card-body d-flex flex-column">
                    <h5 class="card-title">${p.title}</h5>
                    <p class="card-text text-muted small mb-2">${p.short}</p>
                    <div class="mt-auto d-flex justify-content-between align-items-center">
                        <div class="price_tag">€${p.price.toFixed(2)}</div>
                        <button class="btn btn-sm btn-primary add_product" data-id="${p.id}">Add to Cart </button>
                    </div>
                </div>
            </div>` // button creates the connection between product and cart thanks to id, and the col
        product_gallery.appendChild(col);
    });
}

// next few steps are from tutorial https://youtu.be/z_mTYPx49v4?si=PyGte_PX6Q-m8r2H
document.querySelectorAll('.add_product').forEach(btn => {
    btn.addEventListener('click', () => {
        const id = parseInt(btn.dataset.id); // finds in html the str with id and uses it
        add_to_cart(id); // calls the function for this id
    });
}); // finds all buttons with correct key

function add_to_cart(by_id){
    if(cart[by_id]) cart[by_id] +=1;
    else cart[by_id] = 1;
    const btn = document.querySelector(`.add_product[data-id='${by_id}']`);
    if(btn){
        btn.classList.add('added');
        btn.textContent = 'Added to cart';
    }
}
```

JS functions :`your_cart()` - shows items in the cart, lets you change quantity, remove items, `update_cart()` - updates the cart number on the navbar, `update_calculations()` - calculates subtotal, tax (5%), discount (10% if 3+ items), total.

```
// all calculations
function calculations(){
    let subtotal = 0;
    let item_sum = 0;
    for(const [used_id, col_pr] of Object.entries(cart)){
        const id = parseInt(used_id);
        const p = products.find(x=>x.id==id);
        subtotal += p.price * col_pr;
        item_sum += col_pr;
    }
    const tax_num = 0.05;
    let discount = 0;
    if(item_sum >= 3){
        discount = subtotal * 0.10;
    }
    const tax = (subtotal - discount) * tax_num;
    const total = subtotal - discount + tax;
    return {
        subtotal, tax, discount, total, item_sum
    };
}
```

```

// delete one product
cart_body.querySelectorAll('.remove_product').forEach(btn=>{
    btn.addEventListener('click', ()=>{
        const id = parseInt(btn.dataset.id);
        delete cart[id];
        update_cart();
        your_cart();
    });
});

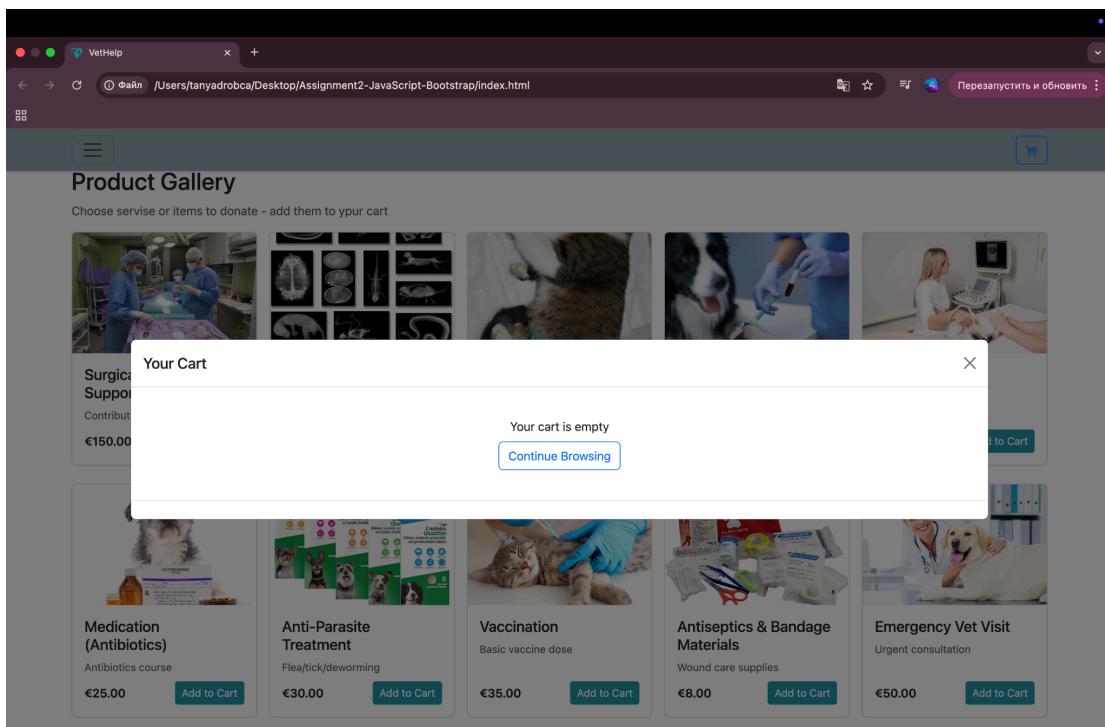
// delit all products
document.getElementById('delit_key').addEventListener('click', ()=>{
    cart={};
    update_cart();
    your_cart();
});

// if i add one more product
document.getElementById('checkout_key').addEventListener('click', ()=>{
    show_form();
});
update_calculations();
}

```

Buttons: Clear Cart (`delit_key`), Checkout (`checkout_key`) - opens checkout form, Remove individual item (`remove_product`). For every button there is a function.

I would like to mention that modal window automatically fits on small screens (`modal-xl`).



```

// create the cart
function your_cart(){
    cart_head.textContent = 'Your Cart';
    cart_body.innerHTML='';
    cart_footer.innerHTML= '';
    if(Object.keys(cart).length === 0){ //if there is nothing in my cart
        cart_body.innerHTML = `
            <div class="text-center py-4">
                <p class="mb-2">Your cart is empty</p>
                <button class="btn btn-outline-primary" data-bs-dismiss="modal">Continue Browsing</button>
            </div>`;
        return;
    }
}

```

If there are zero items in the cart, the body space is automatically filled with the message “Your cart is empty” and the “Return to shopping” button.

## Sources:

- Bootstrap Modals: <https://getbootstrap.com/docs/5.3/components/modal/>
- JS tutorial for updating quantities: [https://youtu.be/z\\_mTYpx49v4?si=PyGte\\_PX6O-m8r2H](https://youtu.be/z_mTYpx49v4?si=PyGte_PX6O-m8r2H)

After completing their shopping, users fill out a **registration form** created by JS (`show_form()`).

Fields: name, phone, address, ZIP, email, and optional notes. The function `form_valid()` checks how correct the entered text is: Name and address - not empty, Phone - digits only (`^\d+$`), ZIP - 1–6 characters, Email - standard email regex (`^[\s@]+@[^\s@]+\.[^\s@]+\$`).

```
let valid = true

if(!name.value.trim()){
    name.classList.add('is-invalid');
    valid = false;
}
else name.classList.remove('is-invalid');

// the same check as for email
if(!(/^\d+$/.test(phone.value.trim()))){
    phone.classList.add('is-invalid');
    valid = false;
}
else phone.classList.remove('is-invalid');

if(!address.value.trim()){
    address.classList.add('is-invalid');
    valid = false;
}
else address.classList.remove('is-invalid');

if(zip.value.trim().length === 0 || zip.value.trim().length > 6){
    zip.classList.add('is-invalid');
    valid = false;
}
else zip.classList.remove('is-invalid');

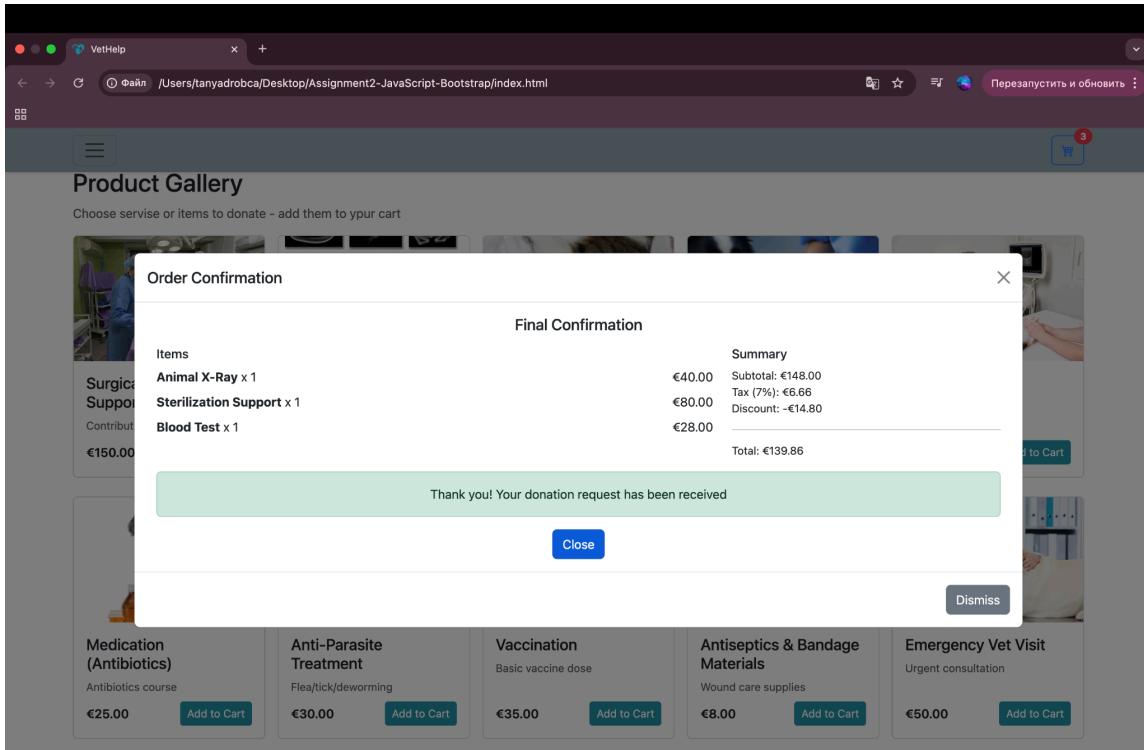
// absatz "Check if a string is an email" https://codebrainer.com/blog/contact-form-in-javascript/
if(!/^\s@+@\s]+\.\s@\s+\$.test(email.value.trim())){
    email.classList.add('is-invalid');
    valid = false;
}
else email.classList.remove('is-invalid');

return valid;
```

Bootstrap grid (`row`, `col-*`) makes the form look good on any screen.

## Sources:

- JS Form validation: <https://codebrainer.com/blog/contact-form-in-javascript>



After submission, `confirm_bill()` displays the **final order summary**: List of items with quantity and individual prices, summary of subtotal, tax (7%), discount, and total, bootstrap alert (`alert-success`) for a success message, close button (`close_resert`) resets cart and hides modal.

Layout uses Bootstrap grid: left column for items, right column for totals.

"We have to speak up for those who cannot speak for themselves." — Peter Singer

**Contacts**  
Email: family.vet881@gmail.com  
Phone: +37369717881  
Address: Șos. Muncesti 94, Chișinău

**Follow us**  
[Instagram](#) [Facebook](#)

  
Our partner clinic

© 2025 VetHelp - Drobca Tatiana's project

Footer contains three columns:

1. Contacts (`text-start`) - email, phone, address of volunteer groups
2. Social media links (`text-center`) - Instagram, Facebook using Bootstrap Icons (`bi bi-instagram`, `bi bi-facebook`)
3. Partner clinic (`text-md-end`) - logo image (`img-fluid rounded`) and description

Columns stack vertically on small screens (`col-12 col-md-4`) thanks to these bootstraps.

The website I created adapts to different devices—for each page, I chose the appropriate Bootstraps so that the text and images fit within the screen and display correctly on different devices.

**Your Cart**

<b>Surgical Operation Support</b> Contribution to surgeo fund	€150.00	1	X
<b>Animal X-Ray</b> Diagnostic radiography	€40.00	1	X

Subtotal: €190.00  
Tax (5%): €9.50  
Discount: €0.00  
**Total: €199.50**

[Checkout](#)

The website I created adapts to different devices—for each page, I chose the appropriate Bootstraps so that the text and images fit within the screen and display correctly on different devices.

In the end of my html code I listed all the Bootstraps I need and their functions—this makes it easier for me to navigate and choose the ones that are suitable for use.

```
118 <!-- bootstraps for me
119 for header https://getbootstrap.com/docs/5.3/components/navbar/ :
120   navbar - nav container
121   navbar-expand-lg - for burger menu on small screens
122   navbar-light - light color
123   shadow-sm - small shadow under nav
124   sticky-top - keeps the nav fixed in top (it doesn't work why?) !css helps
125   navbar-toggler - burger menu
126   navbar-toggler-icon - icon in three lines
127   data-bs-toggle="collapse" - for JS to open and close (thanks to https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js)
128   navbar-nav - container for nav links
129   ms-auto - pushes nav to the right
130   align-items-lg-center - centers items on large screens
131   nav-item - wrapper for nav item
132   nav-link - style for nav link
133 for cart:
134   btn - basic bootstrap button style
135   btn-outline-primary - button with blue outline style
136   ms-2 - adds left margin
137   badge - small label for cart
138   rounded-pill - rounded icon
139   bg-danger - red color
140   position-absolute - positions cart relative to the main content
141   top-0 - aligns element to the top
142   start-100 - moves element to right side
143   translate-middle - centers elements
144 for text:
145   display-5 - text size
146   fw-bold - makes text bold
147   text-center - centers text
148   text-primary - text with primary blue color
149   text-muted - light-gray text color
150   small - smaller font size
151 for main part and gallery:
152   container - centers content
153   row - horizontal layout row
154   col-lg-6 - adaptation for columns on screens
155   gy-4, gx-3 - vertical spacing and horizontal spacing
156   d-flex - flexbox
157
158   justify-content-between - space between items
159   align-items-end - align items to bottom
160   mb-3 - adds bottom margin
161 for logo img:
162   img-fluid - makes img responsive
163   rounded - rounded corners
164 for rooter:
165   bg-light - light background
166   pt-4 - padding on top
167   mt-3 - margin on top
168   mb-0 - removes bottom margin
169 for hidden box with cart https://getbootstrap.com/docs/5.3/components/modal/:
170   modal - creates modal window
171   fade - open/close animation
172   modal-xl - large modal size
173   modal-dialog-centered - the content in center
174   modal-header, modal-body, modal-footer - body content of cart, button and tab section for cart
175 for bootstrap icons:
176   bi - class for bootstrap icons
177   bi-cart4 - cart icon
178   bi-facebook - facebook icon
179   bi-instagram - inst icon
180
```

```
141   position-absolute - positions cart relative to the main content
142   top-0 - aligns element to the top
143   start-100 - moves element to right side
144   translate-middle - centers elements
145 for text:
146   display-5 - text size
147   fw-bold - makes text bold
148   text-center - centers text
149   text-primary - text with primary blue color
150   text-muted - light-gray text color
151   small - smaller font size
152 for main part and gallery:
153   container - centers content
154   row - horizontal layout row
155   col-lg-6 - adaptation for columns on screens
156   gy-4, gx-3 - vertical spacing and horizontal spacing
157   d-flex - flexbox
158   justify-content-between - space between items
159   align-items-end - align items to bottom
160   mb-3 - adds bottom margin
161 for logo img:
162   img-fluid - makes img responsive
163   rounded - rounded corners
164 for rooter:
165   bg-light - light background
166   pt-4 - padding on top
167   mt-3 - margin on top
168   mb-0 - removes bottom margin
169 for hidden box with cart https://getbootstrap.com/docs/5.3/components/modal/:
170   modal - creates modal window
171   fade - open/close animation
172   modal-xl - large modal size
173   modal-dialog-centered - the content in center
174   modal-header, modal-body, modal-footer - body content of cart, button and tab section for cart
175 for bootstrap icons:
176   bi - class for bootstrap icons
177   bi-cart4 - cart icon
178   bi-facebook - facebook icon
179   bi-instagram - inst icon
180
```