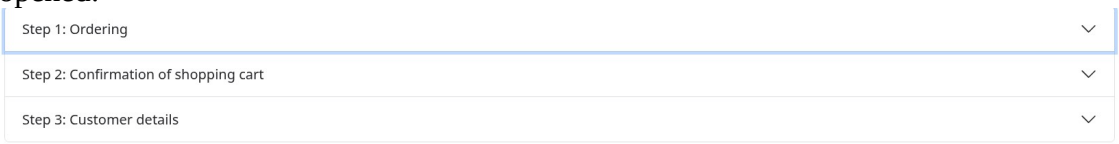
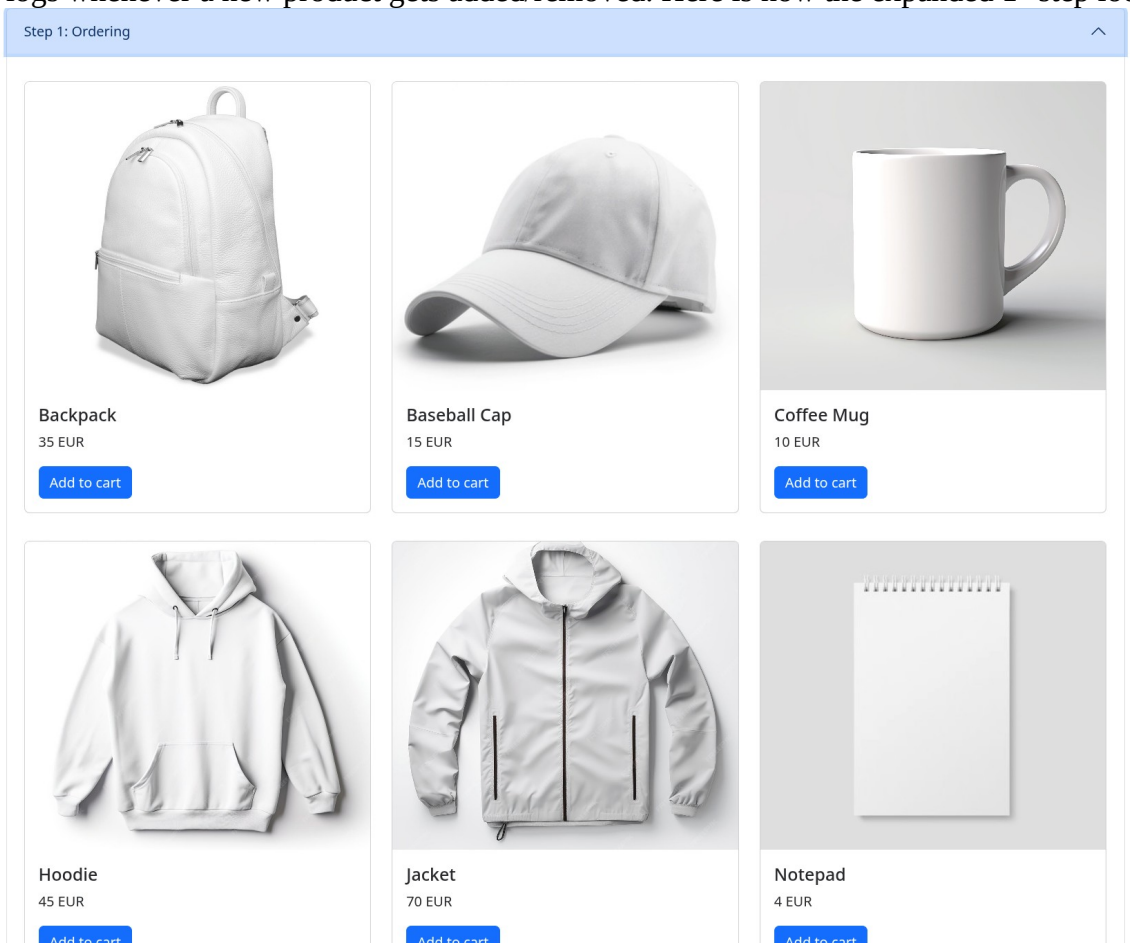
In short: This webpage is a simple implementation of an online shop’s flow. You can add and remove products, review your cart, fill in your details and get a purchase summary.

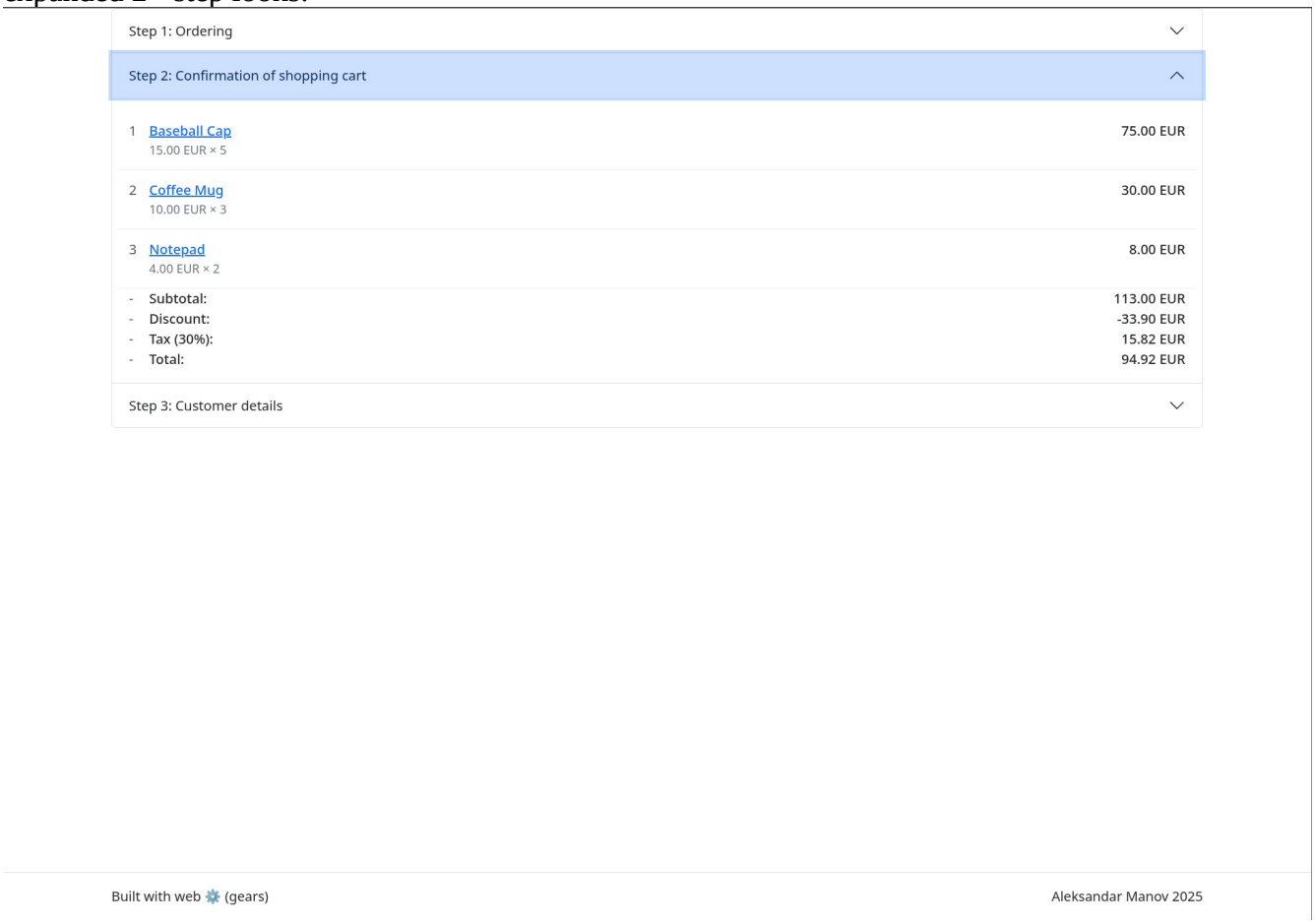
The site is broken down into three steps via the main accordion, here is what the page looks with all the steps unopened:



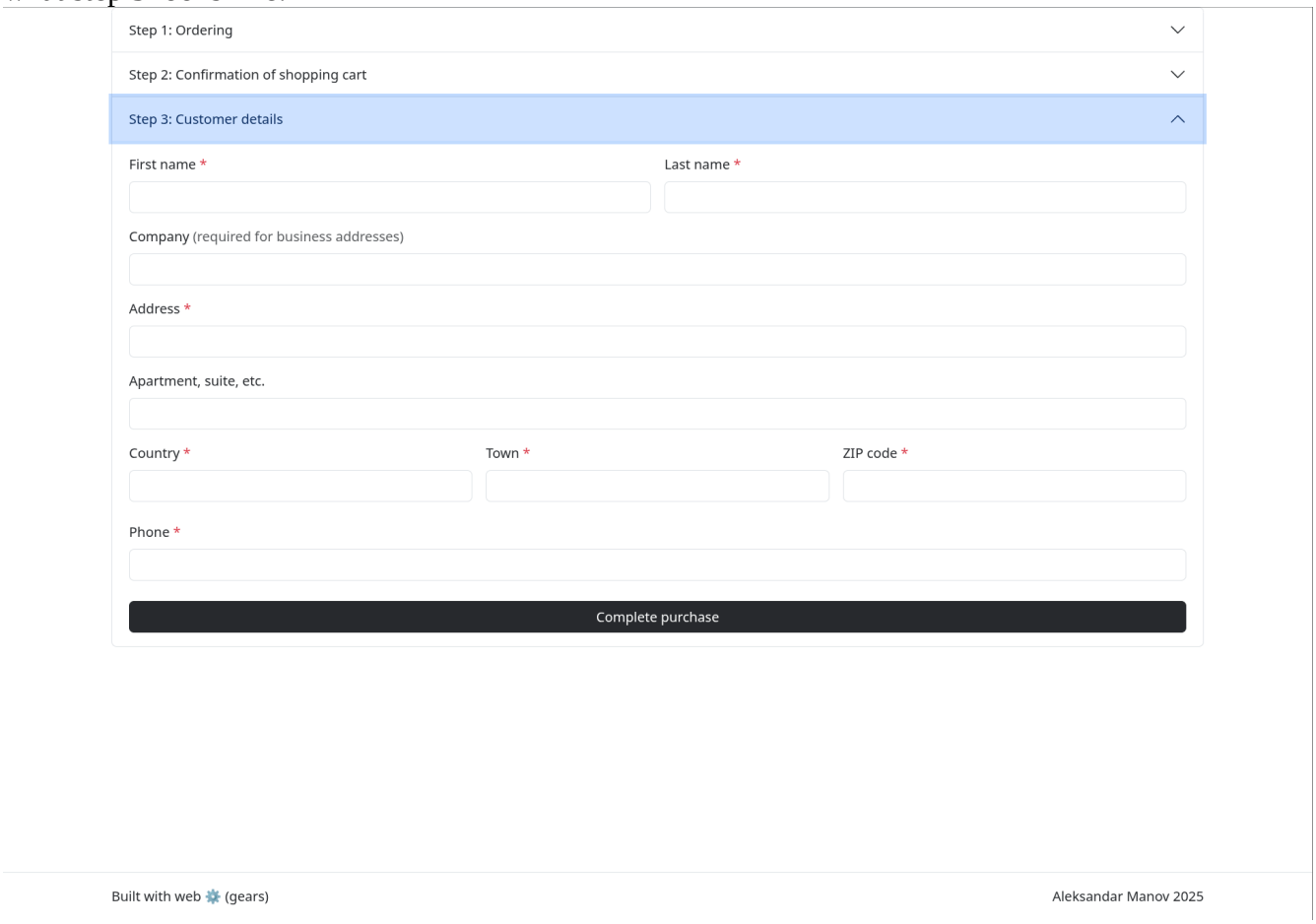
Step 1 is for choosing products. When opening the 1st part of the accordion the user will be presented with a grid of cards with images, buttons and text-like elements to resemble a product, its price and a way to add it. Upon clicking the button there won't be an indication anywhere that the user has added a product into the cart (the user can find that information out in the 2nd step). However the developer console logs whenever a new product gets added/removed. Here is how the expanded 1st step looks:



Step 2 is for reviewing the chosen products from the 1st step, making changes to the shopping cart and reviewing the total, tax, discount and subtotal price. Each selected product is represented via a list where each row has an anchor tag, and a bunch of text-like tags which tell the subtotal price of each product, the name of the product and the original price. Upon clicking one of the anchor tags in the product list, the chosen product will have 1 count of itself removed. Furthermore when there are no selected products the 2nd step will indicate to go back to step 1 to pick products. Here is how the expanded 2nd step looks:



Step 3 is for contact details. It contains a form with a field for customer names, company name, address, apartment, country, town, ZIP code and phone number. When the user unfolds this step before picking any products or whenever the shopping cart is empty the form will be disabled and the user will not be able to input anything into it or submit it. Upon clicking the ‘Complete purchase’ button a modal will pop up which shows the complete shopping cart along with the filled in contact details. Here is what step 3 looks like:



Notes about the JavaScript code:

- The intent was to make step 1 and 2 dynamic. The way I achieved this was by building functions which act as building blocks for laying out the complete DOM, similar to React, however without JSX. The functions in question are all functions whose names are prefixed with ‘generate’. These functions return either one HTML element or an array of them.
- Some of the validation on the form in step 3 is handled by Bootstrap, in particular errors for the ZIP and phone field are shown thanks to Bootstrap’s helper classes.
- Another intent was to avoid using for loops altogether in favor of the ‘pure’ function-programming-like approach methods, such as reduce or map. One exception is in function ‘generateProductGrid’ with no particular reason other than it was simpler to implement like that initially.
- RegEx was used in the validation process and when generating the body for the modal, specifically the contact part.

Notes about Bootstrap:

- The grid system was used for laying out the product catalogue in the 1st step. In particular the responsive classes such as ‘col-md/lg/etc-*’ were used for making the webpage work on different screens.
- Cards were used for laying out each product of the product catalogue in the 1st step.
- Bootstrap’s buttons were used in 1st step for the ‘Add to cart’ buttons and in the 3rd step for ‘Completing the purchase’.
- The close button was used in the confirmation modal.
- The modal was used as a final step of the shopping flow displaying all purchased items and address details.
- The accordion was used for gluing together all 3 steps.
- Helper classes like ‘m*-*’ or ‘p*-*’ were used instead of CSS when spacing elements.