**Modern Frontend Development – Mini Project**

**Problem statement**

So what was the problem? First and foremost I did not have an Angular 2 website, and I had never even touched Angular before, so it was a completely new world that I jumped into.

Furthermore, I had some other requirements that I needed to have implemented into the website. Many of the things were new to me so it obviously was a good fight I had trying to use these different tools and at the same time actually learning them.

So how do I solve these problems and what will the world look like if I do solve them? Let’s take a look at that throughout the report.

**Building the website**

* **Home page**

The header

First things first I started out with the header of the home page of course. Throughout the website I have used Flexbox, which was also new to me, to style almost everything to better be able to put elements on the same row but split off to left and right, which in most cases has been a success.

On the first row I put a logo to the left side and on the row I have phone number, email and the checkout button to the right side. I already loved Flexbox.

Next row was made up by the navigation bar on the left and a search bar to the right, same procedure as the first row.

This was all put together in one header section.

Slider section

The next section was for the slider. I did research and found a slider online which I copied and try to see how to it looked and worked.

It looked good and I checked out the code, Javascript the most, to try to understand how it was build, and I found it quite understandable, using different codes to change for how long the fading should be and how many seconds should pass before the slider showed the next picture etc.

On the slider I put some text with a button below it with it all having a black transparent background.

Categories section

Below the slider we have the categories section which is made up with three boxes. Again Flexbox was useful as I just had to center the children content in the section and then just use justify-content: space-between; to make the boxes split up and go to each direction until 85% of the website which was my default width of my website except for the full width elements.

Products section

In the products section I have two rows, one just for the title “Popular Products” and the other one for the shoes. In the beginning I actually created three rows,

but I realized that even though the shoes are on the rows, I could actually just put them all into one row and then use Flexbox with its flex-flow: row-wrap, which puts all the children content into one row, but pushes the rest of the content below when there is no more space. This way it was also good for mobile responsiveness.

After being able to place the content correctly I started to implement the small details needed such as the red stars above the shoes, the heart and plus to right of the price. The text was simply made up by <h> tags for the title and type and <p> tags for the regular text.

Below it all we have a call to action, a button stating “Buy Now”.

The footer

Last but not least we finally reached the footer which one was of the sections that I was most satisfied with.

The footer is made up by two full width children sections, the top section for the newsletter part and the bottom section for the rest.

In the top section I have one row which includes two children divs, one for the text to the left and one for the input bar and button to the right, again split up with space-between. Wonderful.

Then we have the bottom section which includes info like the addresses of the stores, the latest blog posts, and support.

The bottom section is made up by a row which is made up by three columns. These three columns are just like the categories split up and placed correctly with space-between within the 85% width.

The column to the right is split up in two, a left and right side to be able to have all the text within the same height as the other columns instead of making a long line down.

In conclusion, the homepage was the toughest one for me to build, since many of the things I used were my first time using them. I also restarted with the homepage two times, because of not being satisfied with the way I had put up the elements on the page, and again because I hadn’t used Flexbox to style everything, which meant a less mobile responsive page.

I definitely used most time on the home page, and even though I overall had finished it, I would still come back sometimes to add small details and sharpen the edges different places.

Logically after I had finished it I was also really satisfied with my work when looking at the design of the layout.

* **Detail page**

The header

The next page we have is the detail page, details of the shoes. The first thing we have on this page is of course the header again from the home page. The only thing that changes on different pages on a website is the content. The header and the footer should always stay the same on every page.

The content

My content is made up of one big row in which we have two columns on each side. The left column is for all the shoes/pictures content, and the right column is for the details of the shoes.

The shoe column is then split into two smaller columns, the top one for the big picture, and the bottom one for the small pictures. The pictures in the bottom column are presented in a row.

The detail column is made up of six smaller columns in which we find the product name, brand, price, savings, shoe size, amount, and last but not least the buy button.

The footer

Last but not least we logically have the footer at the bottom of the page again copied directly from the home page after the content is finished.

* **Contact page**

And just because we are good designers with a sense of perfection, we make use of repetition from the C.R.A.P. rules, and therefore we use the same header and footer again on the contact page. Consistency is a must for a professional UI.

Contact form

The content on the contact page meaning the form is simple placed in one section. Since the section is just the whole area between the header and footer, I have also made a div inside the section to target the form itself.

In there we have a div which shows the hasBeenSubmitted text, but it is hidden by default and will only be triggered after the “Send Message” button is clicked.

Since the overall <div> also include all the functions, I have created another div class which targets specifically the box including the input fields and texts.

It is basically a normal HTML and CSS built form with some Angular functions implemented, and since my website has been implemented into Angular, that is what I am going to talk about now.

**Implementing into Angular**

* **The home page**

I made up my home page of five components in Angular.

NewHeaderComponent is made for my header only, so I basically create an HTML file and copy in the header code. All the styling is put into a single file for the whole home page just like you normally do.

To make Angular perceive this header as an component, we have to import { Component } from the core of Angular into our TypeScript file.

The HTML file with the header code needs to be connected to this file which we do by using the templateURL

Last but not least, we then export the class which I named NewHeaderComponent, so we can include it in the app.module.ts file which connects basically everything to each other.

Beside the component for my header, I also created components for the slider, the categories, the products, and of course the footer.

After all these individual components are created, I make a new component which is going to include all the individual component into one component, and that is how all the components are displayed on one single page.

The way to display the HTML in this “father” component is very simple. I basically just include the selectors of the individual components, and all the HTML code from these components will be displayed on the page.

So if the selectors for the header and the footer components are called ‘newheader’ and ‘newfooter’, I simply go to the HTML file of the father component and type <newheader></newheader> and <newfooter></newfooter>, which means that all the HTML code wihin these two components will be used, so you actually do not have to copy the very HTML code itself into the father component.

Basically this is the procedure throughout the three pages.

I actually found it quite easy, and everything is just more fun and exciting when you understand stuff.

The only problems I had with the Angular project were the slider and tabs. It seems that I could not get the Javascript code working in Angular, unfortunately.

**Conclusion**

My conclusion is that I went from not even have touched Angular to actually quite understand it and being able to work in it and implement existing HTML and CSS codes into it.

Furthermore, I have learned the usage of components and the whole workflow of how and why it’s built like this.

I also managed to implement a slider into my project, the one that was not implemented into Angular, and I also managed to create tabs, though I removed it again, as I had difficulties styling it to fit the website theme, and since I’m kind of perfectionistic and have a good understanding of UI and sense for design, I want everything to look good and if it does not, then I don’t want it to be there at all.

Unfortunately, I had these small icons I had to use without functionality for my shoes, and the only reason for that is simply lack of time. If we had had more time, the functionality for those small icons like the stars and the hearts in the shoes part, would not have been an issue for me to build.

Last but not least, I learned to use Flexbox and realized how easy it was and how it made the responsiveness much better, which is also one of the reasons I really liked this project, as I really learned many new things that I can take with me and use in future projects.

Overall, I personally feel this project has been a success for me, learning a lot of new things, everything from learning what Angular is and using it, styling with Flexbox, to learn a new workflow for HTML and using a different approach the setting up of the code.

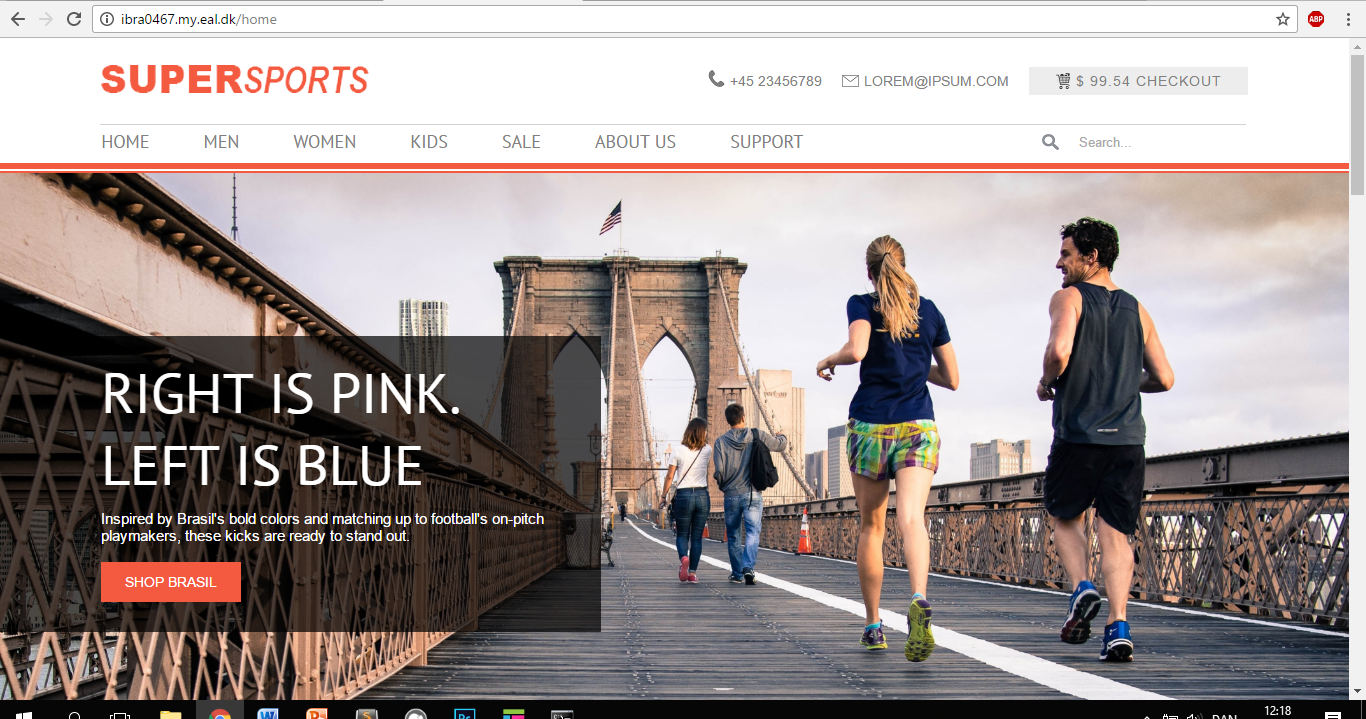
I even learned how to use SASS even though I didn’t use it in my project simply because I had styled almost everything on beforehand and was under time-pressure and I felt it was more important to build the layout of my website first, so I had something to actually show.

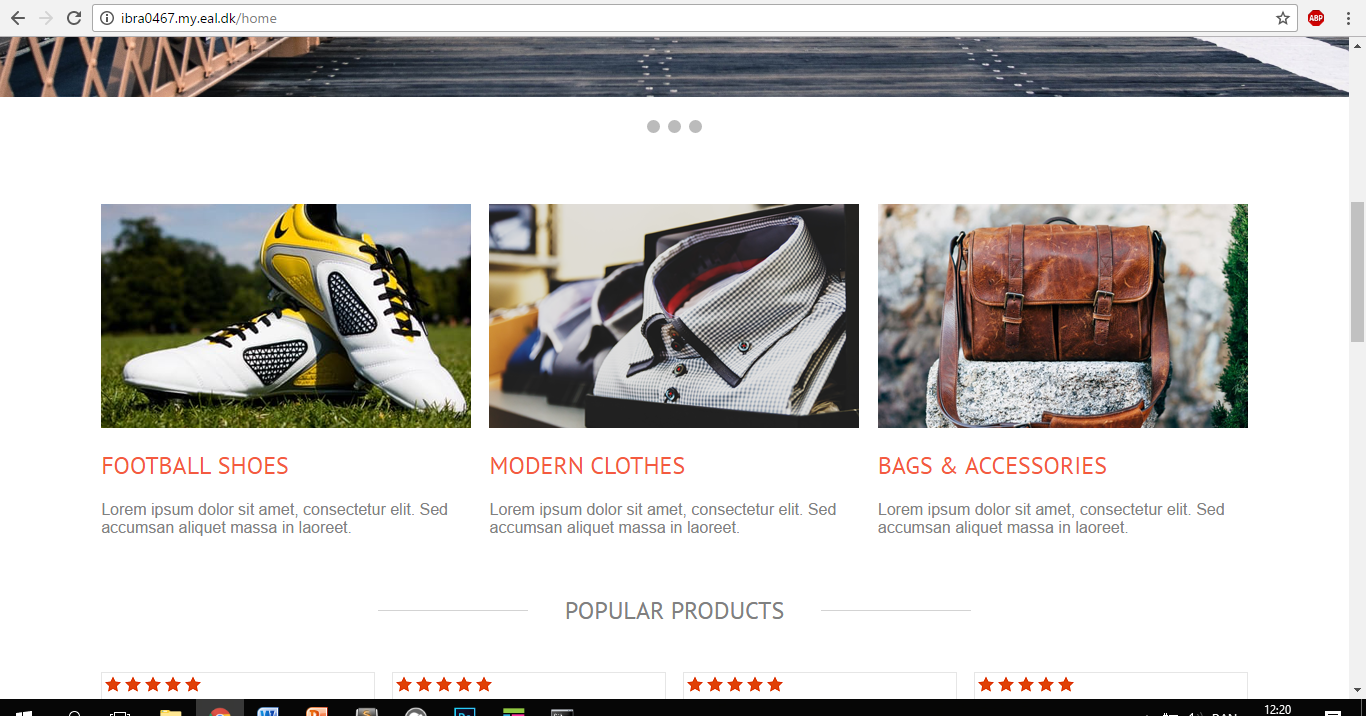
I like how SASS really shortens the styling time with its variables, mixins, inheritance etc. So you can basically re-use code whenever you need it instead of having to write all the code yourself every time you need it. It’s a bit like writing PHP in Object-Oriented Programming.

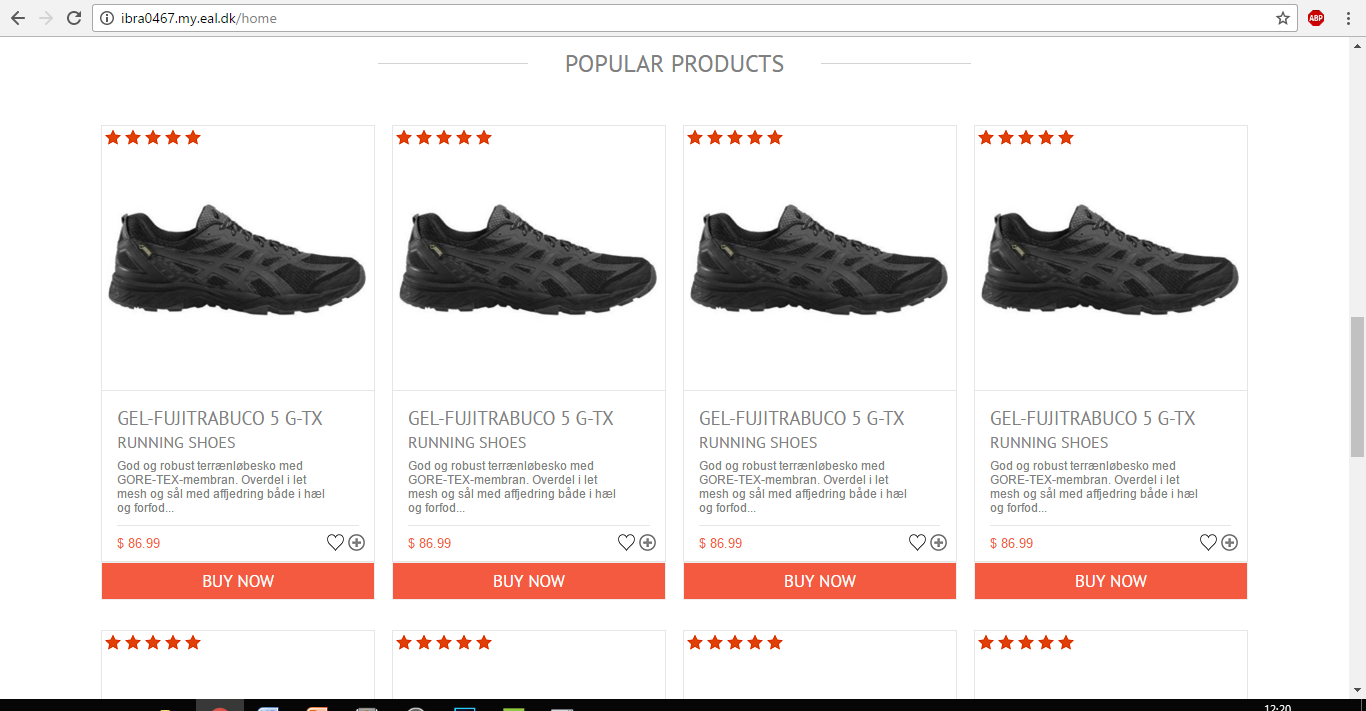
I really encountered many problems throughout my website building, and when you have to code and solve problems at the same time, it takes a lot of time.

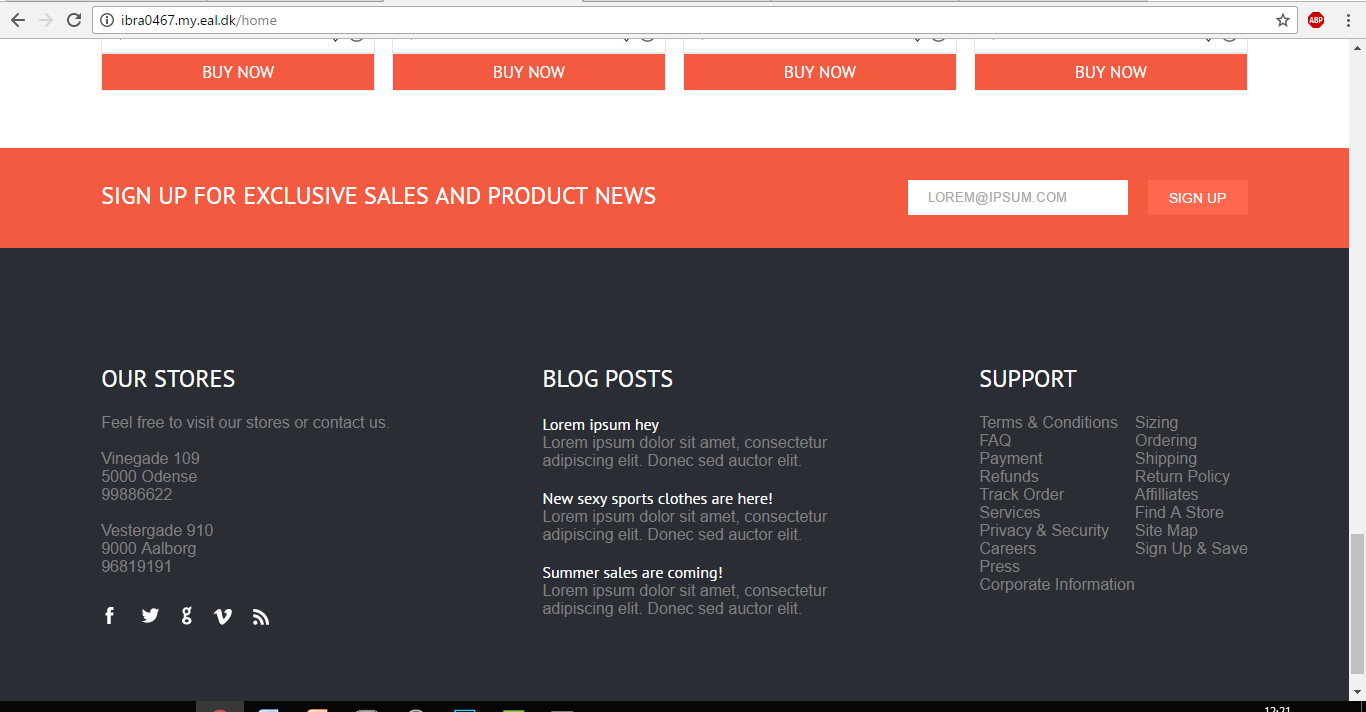
Fortunately, I managed to solve almost all of them which gave me a really satisfying feeling.

HOME PAGE

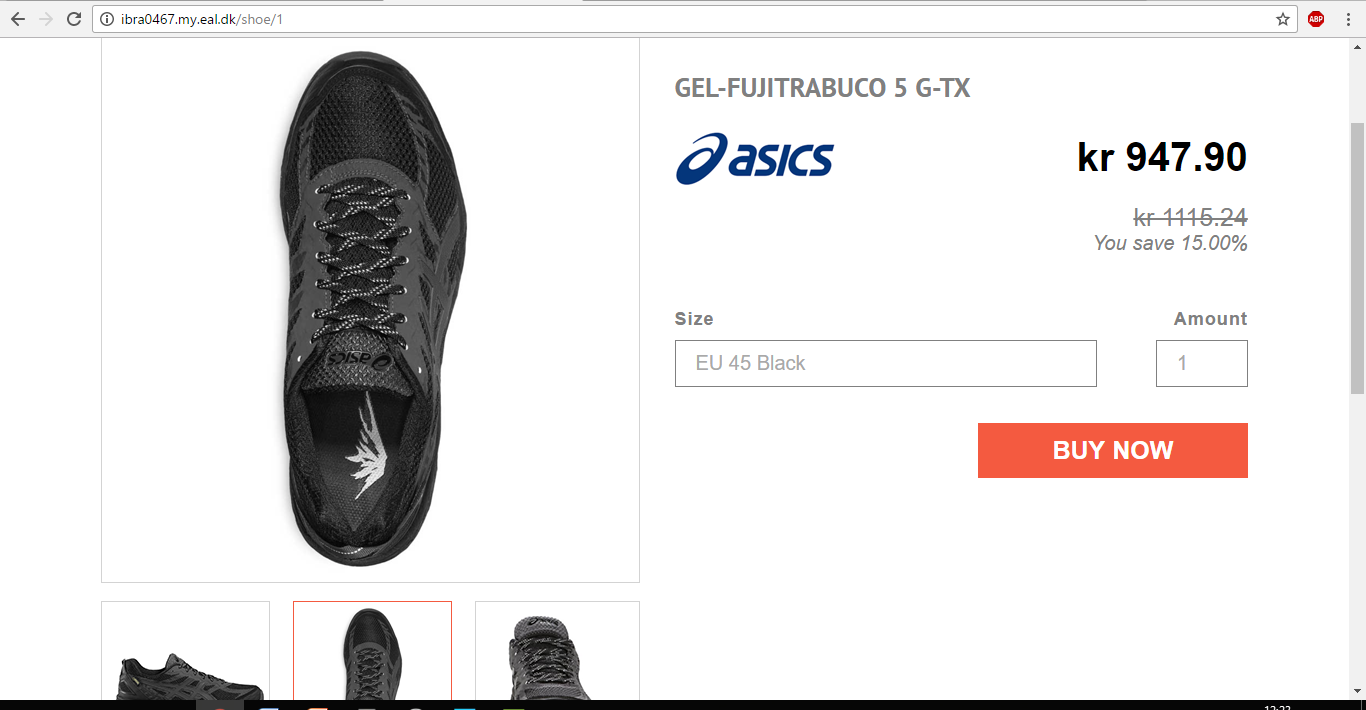








DETAIL PAGE



CONTACT PAGE

