CSS Cheat Sheet

This simple site, build only using HTML5 and CSS, was one of my first websites. You never have to cheat again.

Idea: The assignment was to build your own Cheat Sheet using HTML tables and a customized color scheme.

Approach: As I am more visually focused I chose to add an illustration as a banner. The color scheme was adjusted to this illustration.

Challenges: As it was my first time working with a banner background image, I had some issues getting it to stay centered and not repeating itself in larger screens.

Deventer Running Club

My first responsive website for an imaginative running group. I’ve used CSS flexbox and a bit of CSS Grid to set it up.

Idea: This page was made as a ‘challenge project’ for the front-end engineer path of Codecademy. The assignment was to build a responsive website for a club group using relative units, percentages and media queries.

Approach: I decided to build one for a running group that does not exist (yet) in Deventer. CSS flexbox was used to set up the page and one section uses CSS grid. Next to that there are several media query break points.

Challenges: My first version of this website was responsive, but the use of many breaking points meant that transitions were not fluid. In the second version of this website I have used more relative size units and percentages to indicate width/heights etc. This highly improved the transitions.

Room to improve: Many illustrations used for the page are relatively large. To improve the loading time of the website these can be made smaller. Next to that the site still has room to improve aesthetically.

Find your hat

This interactive terminal game was build with JaveScript Classes. You can read the code on my Github page. Find you hat, but beware of the holes!

Idea: This game was made as a ‘challenge project’for the front-end engineer path of Codecademy. The assignment was to build an interactive terminal game with JaveScript classes and using user input in Node.js.

Approach: The player sees a field with a few holes and one hat. The goal is to find your hat using your keyboard.

Challenges: This whole assignment was pretty challenging as it was the first time the assignment only included functionalities and I had to come up with the necessary functions and how they would interact by myself. Once I had the game working the last challenge was to ensure that the console would not reprint the whole field every time you made a move. With the help of some co-students at the Codecademy forum I added code to ensure the player would keep seeing the same field with only his/her move updated.

Room to improve: This game can be made more exiting by adding more rows and columns. I would also like to make it possible to play this game online. Therefore I would like to rebuild this game using React to ensure the field would not have to be reloaded after each move.

Eighties workouts

This personal project let me practice my web development skills as well as my dance moves.

Idea: This idea came from a personal frustration, as I couldn’t find good quality eighties workout video’s online. I decided to build my own website to create on easy place from which you can link to workouts.

Approach: This site continues to grow together with my skills. The first version was build with some basic HTML and CSS knowledge. Over time I created version 2 (responsive) and version 3 (new design and improved responsiveness).

Challenges: The main challenge of this website was deciding on the design and color scheme. The first version was completely purple. I decided to tone this down (a bit) in the third version to leave more room for the content.

Room to improve: There are many ideas to improve this site with more personal content and of course more workout video’s.

Jammming with MariaBee

The webapp was build using React and the Spotify API. Create you own playlist and save it in your Spotify account.

Idea: This webapp was made as a ‘challenge project’ for the front-end engineer path of Codecademy. The assignment was to make a React web application called Jammming in which you would use your knowledge of React components, passing state and requests with the Spotify API. The app should enable users to search for songs in the search bar, see the results of the search and add songs to a ‘new playlist’. In the new playlist you can delete songs and once your list had the perfect tunes, save it in your Spotify account.

Approach: The assignment was divided in 13 sections, each section containing steps like ‘Pass the playlist tracks from the ‘Playlist’component to the ‘Tracklist’ component. This helped in the set up of the app and the way the components should interact. It still left it up to the student to write the correct code.

Once the app was running as it should, I build an extra feature to organize search results based on popularity.

Challenges: The interaction between different React components is a challenge I found incredibly interesting, but also difficult, especially in combining this with the correct syntax. This project took me several days to finish.

Room to improve: there are many ideas for new features in this app. However, it made me realize that more than improving this specific app, I want to follow more courses on React and build more apps to further improve my React skills.