Eigenreflexion

Bevor meine Gruppe das Arbeiten begonnen hat, haben wir uns alle selbstständig nochmal Stencil angeschaut: <https://www.youtube.com/watch?v=MqMYaT1GlWY>, <https://www.youtube.com/watch?v=SCB3X2ApYWc>.

Das hat mir sehr geholfen, da ich am Anfang erstmal wenig verstanden habe. Nachdem ich mir aber nochmal näher angeschaut habe, wie Stencil funktioniert, habe ich schnell gemerkt, dass es gar nicht so schwer ist, wie ich im ersten Moment dachte. Unsere Gruppe hat sich dann getroffen und ein grobes Scribble unserer Startwebseite und den Unterwebseiten erstellt. Dabei haben wir auch direkt die verschiedenen Components definiert. Im nächsten Zug haben wir die Components unter den Mitgliedern aufgeteilt. Ich wollte gerne die Navigations-Bar machen und mich dafür entschieden. Außerdem haben wir darauf geachtet, dass jeder einen Component hat, der etwas anspruchsvoller ist und einen der leichter ist. Deshalb habe ich noch unseren „my-text“ Component und den „my-welcometext“ erstellt. Außerdem haben wir eine Schriftart und verschiedene Farben festgelegt, damit wir alle mit denselben arbeiten. Auf das genaue Design sind wir dann aber erst später eingegangen. Jeder war außerdem für den Inhalt von zwei Seiten zuständig. Ich habe hierbei die Seite Ähnlichkeit und Schicksal gemacht.

My-text

Für diesen Component habe ich mir einen Ordner mit der tsx und css Datei erstellt. Für den Component brauche ich zunächst eine Property, die ein String ist, damit auf jeder Seite ein anderer Text eingefügt werden kann. Jedoch hat Marco mit dir hier nochmal etwas verändert, deshalb ist hier jetzt ein Boolean definiert. Danach erstelle ich als Component ein einfaches p-Element, das als Inhalt die davor definierte Property enthält. Auch diesen hat Marco mit dir noch einmal angepasst, da wir eigentlich Slots verwenden wollten. In der CSS-Datei habe ich nur die Schriftart angepasst und text-align. Dieser Component wir später dann in der „flexdiv“ verwendet.

My-welcomeText

Dafür habe ich zunächst den „My-text“ Component kopiert. Hier kannst du eigentlich auch nochmal sehen, wie der Component „my-text“ eigentlich aussah. Hier habe ich aber ein div verwendet, damit der Absatz besser zu stylen ist. Danach habe ich das Sytling so gestaltet, dass ich unsere ausgewählte Schriftart angepasst habe und der Absatz mittig auf der Seite erscheint, in einem Block erscheint.

My-nav bar

Da diese auf jeder Seite gleich sein soll und die Inhalte nicht angepasst werden sollen, habe ich keine Properties verwendet. Zunächst habe ich also eine ganz normale Navigationsbar über eine Liste erstellt und auch bereits ein Sytling gestaltet. Ich habe mich als erstes für schwarz als Hintergrundfarbe und weiße Schrift entschieden, die beim hovern über die einzelnen Elemente rot wird. In der Navigationsbar ist für mich dann das Problem aufgetreten, dass wenn man bereits auf einer Unterseite ist nicht mehr „weiterkommt“, da der hinterlegte Link dann nicht mehr passt. Also habe ich eine If-Funtkion erstellt. Diese sah zunächst folgendermaßen aus.

export class MyNav {

render() {

if (window.location.href=="html"){

    return <header>

      <nav>

        <ul class="navbar">

          <li><a href="html">Start</a></li>

          <li><a href="html\aehnlichkeit.html">Ähnlichkeit</a></li>

          <li><a href="html\geschlossenheit.html">Geschlossenheit</a></li>

          <li><a href="html\fortsetzung.html">Gute Fortsetzung</a></li>

          <li><a href="html\naehe.html">Nähe</a></li>

          <li><a href="html\praegnanz.html">Prägnanz</a></li>

          <li><a href="html\schicksal.html">Schicksal</a></li>

        </ul>

      </nav>

    </header>;

    }else{

      return <header>

      <nav>

        <ul class="navbar">

          <li><a href="..\html">Start</a></li>

          <li><a href="..\html\aehnlichkeit.html">Ähnlichkeit</a></li>

          <li><a href="..\html\geschlossenheit.html">Geschlossenheit</a></li>

          <li><a href="..\html\fortsetzung.html">Gute Fortsetzung</a></li>

          <li><a href="..\html\naehe.html">Nähe</a></li>

          <li><a href="..\html\praegnanz.html">Prägnanz</a></li>

          <li><a href="..\html\schicksal.html">Schicksal</a></li>

        </ul>

      </nav>

    </header>;

    }}

}

Ich habe in „If“ abgefragt, ob ich mich auf der Startseite befinde und dann dafür die Links wie ich sie zuvor hatte ausgegeben. Wenn ich mich aber nicht auf der Startseite befinde, soll der Link erst nochmal einen Schritt zurück. So können wir auch von jeder Unterseite eine andere aufrufen und gleichzeitig wird in der URL nicht der ganze Weg angezeigt, sondern nur die Startseite und die aktuelle Unterseite. Dadurch wird diese nicht extrem lang, wenn der User zwischen den Seiten wechselt und ist immer übersichtlich.

Christian hat mir dann noch den Tipp gegeben, dass ich einer Variable auch verschiedene Werte zuweisen kann. Dadurch konnte ich den Code nochmal kürzen, wie er dann auch aktuell im Script zu sehen ist. Dazu habe ich eine Variable „link“ erstellt, die in der If-Funktion den Anfang des Links enthält. Wenn ich mich also auf der Startseite befinde, ist die Variable leer auf einer Unterseite enthält sie aber „einszurück“. Dadurch kann ich danach nur einmal die Ausgabe der Navigtionsbar schreiben und vor den Link die Variable setzen.

Damit habe ich auch nochmal Marco geholfen und wir haben im Footer für das Impressum das gleiche Prinzip angewendet.

Eigenreflexion

Nachdem wir alle unsere Components erstellt hatten, haben wir uns nochmal genauer über das Design unterhalten und das bereits vorhandene angeschaut. Wir haben uns dann abgestimmt und das Design vollendet. Zuerst wollten wir ein Template erstellen. Marco hat mit dir dann aber abgeklärt, dass wir es auch ohne machen können. Dann hat jeder die Seiten für die er zuständig war mithilfe der Components aufgebaut. Bevor ich mich mit dem Thema Stencil genauer auseinandergesetzt habe, kam mir die Aufgabe sehr schwer und erstmal unübersichtlich vor. Im Nachhinein kann ich aber sagen es ist einfacher als gedacht. Von allen drei Assignments viel mir dieses am leichtesten. Hier war eine wichtige Rolle unsere Zusammenarbeit, die sehr gut funktioniert hat. Wir haben uns gegenseitig unterstützt und auch auf kleine Fehler oder Verbesserungen hingewiesen, wodurch jeder von uns nochmal etwas dazugelernt hat.