

Projektopgave - Del 6

Github url: <https://github.com/larsk7cdk/vejrportalen>

Branch: del-6

Vejrportalen url: <https://40.127.170.50/vejrportalen/>

Opgavebeskrivelse

Tilpas dit projekt ud fra det, du har lært i denne uge.

1. Du skal i en tråd vise før og efter billeder (screenshots). Forklar kort hvad du har gjort, og hvorfor du har gjort det.

2. Kommenter på en af dine medstuderende, der har lavet ændringer, du synes godt om

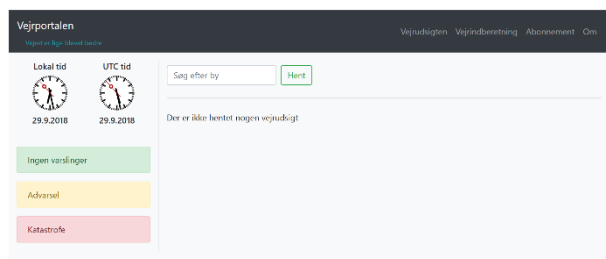
Indledning

I del-6 af projektopgaven som drejer sig om design, har mit fokus været på at få gjort forsiden af vejrportalen mere simpel. Jeg har taget nogle af design principper fra lærebogen og benyttet dem på vejrportalen.

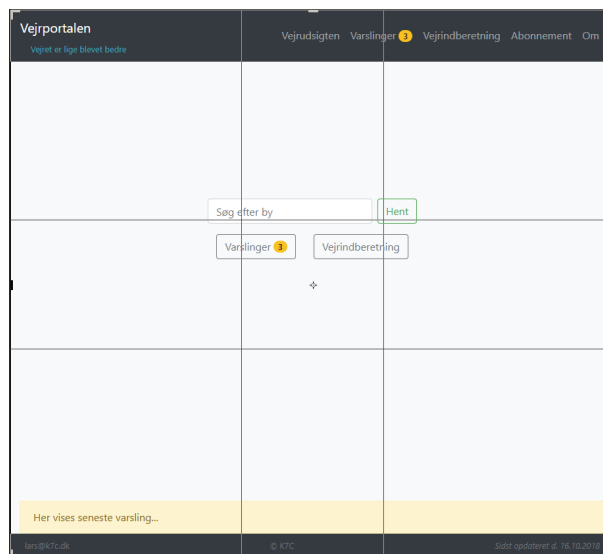
Derudover har jeg tilføjet en funktion der indlæser vejrdato for den placering man befinder sig på, hvis placering er tilladt. Er dette ikke tilfældet, vil søgefeltet vises.

Vejrudsigten

Følgende er vejraportalens forside i et før og efter billede, hvor placering ikke er tilladt. Hvis placering er tilladt, vil vejrdato fremsøges med det samme.



figur 1 – Vejruddigten før



figur 2 – Vejruddigten efter

For at få brugeren til at fokusere på det som er vigtigt, har jeg valgt at fjerne urene og varslinger til sin egen side. Dermed er der ingen forstyrrende elementer. Dog er det muligt hurtigt at kunne komme til siden, der giver et overblik over aktive varslinger. Der er ligeledes en genvej til vejruddberetning, da jeg mener de har relevans på forsiden.

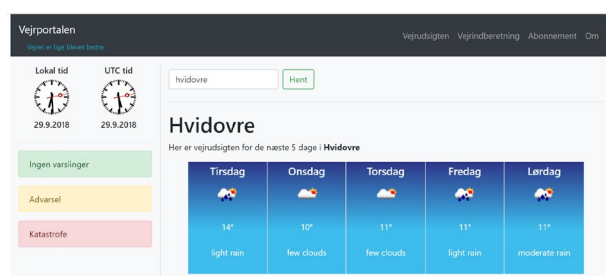
For at vise antallet af aktive varslinger, er dette placeret som et tal både på genvejen og i menu feltet.

Jeg har placeret søgefeltet så det for så vidt muligt rammer det gyldne snit.

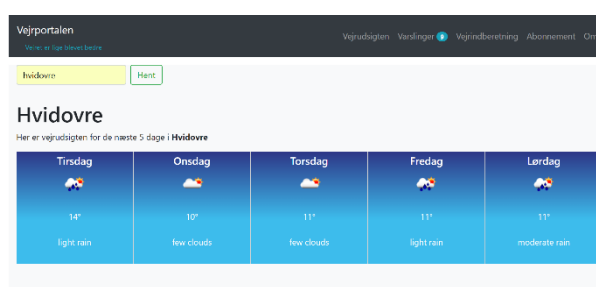
Header'en er gengivet på alle sider, for at give genkendelighed for brugeren. Dette gør ligegyldigt hvilken side man befinder sig på, ved man det er vejraportalen.

Vejrdato

Visning af vejrdato er ændret til at benytte hele vinduets bredde. Siden er responsiv og vil på mobil stække kortene så de står på rækker. Dette giver en mere ensartethed på siden. Igen har fokus været på kun at vise data som er relevante for brugeren.



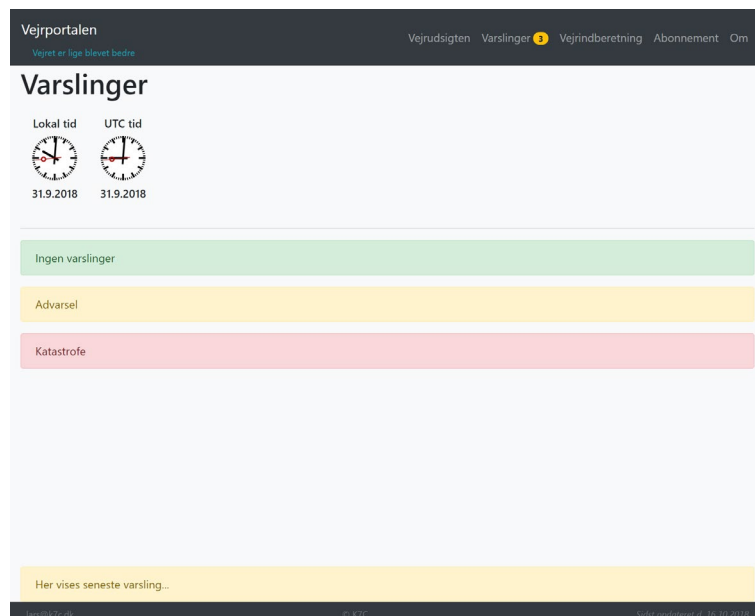
figur 3 – Vejrdato før



figur 4 – Vejrdato efter

Varslinger

Varslinger er flyttet til sin egen side som ses på figur 5, og urene er placeret over varslinger, således man kan se hvor "gammel" en varsling er.



figur 5 – Varslinger

Placering

På figur 6 vises koden der benyttes til at fremfinde bynavn ud fra ens placering, hvis der er givet tilladelse i browseren.

Som det første, hentes længde og breddegrader ved hjælp af HTML5 navigator API. Er det muligt at hente koordinaterne, kaldes videre til et 3. parts API, som henter oplysninger om placeringen. Det der er brug for, er bynavn for at kunne hente vejrdata, hvilket derfor er det eneste som returneres i kaldet.

```
1  var Geo = {};  
2  
3  $(document).ready(function() {  
4      const API_URL = "https://eu1.locationiq.com/v1/";  
5      const API_KEY = "6147b594e590c1";  
6      const API_FORMAT = "json";  
7  
8      Geo.getTown = function() {  
9          return new Promise(function(resolve, reject) {  
10             if (navigator.geolocation) {  
11                 navigator.geolocation.getCurrentPosition(  
12                     function(pos) {  
13                         resolve(pos);  
14                     },  
15                     function(err) {  
16                         reject(err);  
17                     }  
18                 );  
19             }  
20             }).then(function(pos) {  
21                 return new Promise(function(resolve, reject) {  
22                     const url =  
23                         API_URL +  
24                         "reverse.php?key=" +  
25                         API_KEY +  
26                         "&lat=" +  
27                         pos.coords.latitude +  
28                         "&lon=" +  
29                         pos.coords.longitude +  
30                         "&format=" +  
31                         API_FORMAT;  
32  
33                     $.getJSON(url, function(data) {  
34                         resolve(data.address.town);  
35                     }).fail(function(err) {  
36                         reject(err);  
37                     });  
38                 });  
39             });  
40         });  
41     });  
42 }
```

figur 6 – Kode til at hente bynavn ud fra placering