

Rapport 134

KandidatNr: 253

KandidatNr: 249

KandidatNr: 256

Index.html

Hjemmesiden er blitt laget med en enkel layout hvor det eneste innholdet er en tekst på siden og en navigasjonsbar i headeren, som enkelt lar bruker gå videre til lokasjon siden. Hele siden er blitt lagt i flexbox som gjør at siden er responsiv og har muligheten til å tilpasse seg med tanke på hvilken enhet siden blir brukt på.

Lokasjoner.html

Denne siden fungerer som en oppslagsside og knytter alle sidene sammen. Hver av sidene har en link som viser til lokasjoner.html og det gjør at den fungerer som et knutepunkt. Utover det så har ikke denne siden noen spesielle funksjoner.

Bryggen.html

Er satt opp med et kart som har en markør som viser hvor bryggen ligger, samt en tekst som kort beskriver bryggen i Bergen. Kartet lastes når siden åpnes og markøren er lagt inn på forhånd, siden den aldri vil endres.

Toalettkart.html

Dette er siden der vi henter dokart.json filen i fra difi. Siden blir initialisert av loadJSON() funksjonen. Funksjonen sjekker hvilken side han befinner seg på når han kjører, og når han er på toalettkart.html, så vil han kalle på initTable(). initTable() ser på datasettet og legger hvert element inn i et listeelement. Den kaller så på initMap2(), for å laste inn kart fra google api. Når kartet bli initialisert så starter han en "for"-løkke som skal plassere markørene på kartet. Her bruker vi en funksjon som heter pos(), til å sammenligne navnet til elementet i listen med navnet til en plassering i datasettet. Om det er en match så returnerer han et objekt med lat,long koordinater og returnerer dette til bruk i initMap2().

På siden så har vi tre søkefunksjoner. Checkbokser, vanlig søk på navn og hurtigsøk, som fungerer som checkbokser i tekstform. Vanlig navnesøk reagerer på onkeyup, og kaller filterTable(). FilterTable() ser på om det som er skrevet inn matcher noen av listeelementene og fjerner fortløpende de som ikke matcher. Etter søk må man refreshe siden eller klikke på reset knappen, for å få listen tilbake.

Hurtigsøket fungerer slik at man skriver inn en string som matcher en regex streng. Syntaksen er: "kjønn:herre", "stellerom:ja" eller "ÅpentKl:13:45". Når brukeren har skrevet inn en streng så klikker han på "hurtigsøk" - knappen. Den kjører scan() funksjonen som tar inputen og legger det inn i et søkeobjekt. Om strengen matcher regex så vil en av egenskapene til objektet endre seg. scan(), kaller så på mainSok(), som er hoved søkefunksjonen vår som blir brukt av hurtigsøk og checkboksene. Den består av flere if-setninger som sjekker om objektet sok sine egenskaper matcher noen av de som er listet opp. Når mainSok(), har funnet en egenskap som matcher så starter det en "for"-løkke som ser på om navnet i listen matcher navnet i datasettet og om det elementet i datasettet har eller ikke har den egenskapen vi ser etter. Om elementet i datasettet ikke har den egenskapen vi ser etter så sletter vi den i fra listen og setter iterator verdi "i", til "i=0". Dette er slik av vi ikke skal hoppe over noen elementer når listen blir mindre pga sletting. Når listen består av elementer som har de egenskapene vi har søkt etter så kaller vi på initMap2() igjen for å oppdatere kartet slik at det matcher den nye listen.

Checkboksene fungerer på ganske lik måte som hurtigsøket. Når en checkboks blir merket så blir checkSok() funksjonen kjørt. Denne funksjonen endrer søke-objektet brukt i hurtigsøk, basert på hvilke knapp som blir merket. Om man klikker på "kvinne"-knappen så tilsvarer det å skrive "kjønn:dame". Når søke-objektet er endret så kaller vi mainSok() igjen og den samme operasjonen som forklart i hurtigsøk kjører.

Man kan få kontakt med datasettet ved å skrive data.entries i console

Lekeplass.html

Lekeplass fungerer likt som toalett kart. Denne siden blir initialisert med en funksjon som heter loadLekeplass() som henter json filen for lekeplasser i Bergen. Deretter blir funksjonene initLekeplassTable() og initMap3() kjørt. Resten fungerer på samme måte som på toalett kart.

Til slutt vil siden ha et kart med markører som viser posisjonen til hver enkelt lekeplass og ha en nummerert liste med navn på lekeplassene.

Man kan få tilgang til datasettet via console ved å skrive lekeplass.entries

Rom_uib.html

Denne siden inneholder to filtrerings bokser, der man først vil velge hvilket fakultet man er ute etter. Når dette blir valgt lastes en liste over alle rom som tilhører fakultetene som er valgt. Dette blir gjort ved initTableUib() funksjonen som sjekker om "areaname" egenskapen i datasettet samsvarer med det fakultetet som brukeren har merket av. De elementene i datasettet som samsvarer, blir lagt inn i en liste med navn, romNr og en link som består av

en kort oppsummering av rommet. Deretter har man mulighet til å sortere rommene etter navn alfabetisk som blir gjort via funksjonen `sorterEtterNavn()`, som legger alle rom inn i et array og sletter dem fra lister, deretter blir de sortert i arrayet og returnert til listen. Så har vi `sorterRom()` funksjonen som sjekker om rom har "type" egenskap med verdi auditorium, lesesal, møterom eller seminarrom. Her bruker vi `romNr` til å matche liste elementene med elementene fra datasettet siden mangen rom har like navn på tvers av de ulike fakultetene. Deretter vil alle som ikke har "rom typen" som er merket av i checkboksen bli slettet. Vi har også en enkel reset funksjon som laster siden på nytt for å fjerne data som ligger i listen.

Vi har opplevd en feil i rom sorteringen. Om man velger flere enn 1 fakultet så vil ikke filtreringen fungere maksimalt. Velger man for eksempel 2 fakultet og filtrerer på auditorium så vil det første rommet i listen ikke være et auditorium

Man kan få kontakt med datasettet fra console ved å skrive `uib.data`

Favoritt_toalett.html

Denne siden har en dropdown meny som lar bruker velge et spesifikt(favoritt) toalett, når dette blir gjort kjøres `avstand()` funksjonen som sjekker avstand mellom det valgte toalettet og alle lekeplasser i datasettet. Den stopper deretter når den har funnet den laveste verdien. Vi baserte først funksjonen vår på standard Pythagoras, men når vi skulle sammenligne vårt resultat med koordinat kalkulator på nett så var det stort sprik mellom vårt og deres resultat. Vi fant andre måter å gjøre dette på, men bestemte oss for å basere funksjonen på Spherical law of Cosines, med noen endringer.

Css og Javascript oppbygging

Vi har valgt å ha en fil for javascript og en fil for css istedenfor å ha de spredd over flere filer, siden vi alle syns dette var lettere for oss. Å bare ha et stylesheet og et script som lastes gjorde at vi ikke hadde noe behov for å sjekke at det var lagt inn riktig i hver av sidene. Det hindret oss også fra å legge inn feil css eller js kode i en fil der det ikke hørte til. Det kan tenkes at det hadde vært mer ryddig å hatt det delt opp, men vi følte at det var best å ha alt samlet. Å ha det samlet gjorde det også lettere med tanke på media queries da de blir brukt på alle sidene.