INFO134 Semesteroppgave Rapport

Antall gruppemedlem: 1.
Kandidatnummer: 243

HTML sider

index.html
Fremside. Siden linker til alle de andre HTML sidene.

toaletter.html
Denne siden viser en oversikt over toaletter i Bergen. Støtter avansert søk og hurtig søk.

lekeplasser.html
Denne siden viser en oversikt over lekeplasser i Bergen.

favoritt.html
Denne siden lar brukeren velge sitt favoritt toalett. Siden vil så vise det nærmeste toalettet fra valget til brukeren.

custom.html (Oppgavedel 10).
Dette er siden som presenterer det vilkårlige datasettet. (Oppgavedel 10).
Denne siden viser ulike kakediagram som omhandler episoder av fjernsynsprogrammet Norge Rundt. (LES TEKSTEN VED SLUTTEN AV RAPPORTEN)

Her er en link til dette datasettet:
https://hotell.difi.no/?dataset=nrk/norge-rundt

Det er mulig å spesifisere ulike parametere når en vil hente data fra dette datasettet.
Som et eksempel er dette et utdrag for koden jeg skrev i den tilhørende skript filen - «custom.js»:
var aar = «2016».
var url = «https://hotell.difi.no/api/json/nrk/norge-rundt?aar="+aar+"&kommune=Bergen»
Her er jeg bare interessert i episoder der kommunen er «Bergen», og selve året til episoden er en variabel, noe som betyr at vi kan få ulike datasett dersom vi bare forandrer verdien til år-variabelen.

CSS/JS
Hver html-fil har en egen css og javascript fil som er tilknyttet den. De vil ha akkurat samme navn som html-filen de tilhører. «custom.html» linker også til en ekstra javascript fil kalt «canvasjs.min.js» som blir brukt for å presentere en graf på denne siden.

Redegjøring av oppgavedeler

Her vil jeg gå gjennom alle oppgavene delene og forklare litt om funksjonaliteten, valgene jeg gjorde og kanskje mulige forbedringer.

Oppgave 1 CSS/Flexbox/Responsive design

Jeg vektla å bruke mer tid på programmeringsdelene enn på selve utseende til sidene. Derfor er styilngene på sidene ganske simple og ukompliserte. Jeg fokuserte mer på å få sidene til å se enkle og oversiktlige når det gjelder presentasjon av kode, enn at de ser ut som «normale» proffosjonelle nettsider. Flexbox har vært svært nyttig for å gruppere elementer i rader, kolonner eller å sentrer dem, eller lage symmetriske mellom rom mellom. Det er hovesaklig dette jeg har brukt FlexBox for.

Oppgave 2 JSON-funksjon

Jeg har laget en funksjon som henter JSON via et HTTPRequest basert på callbacks. Den er kalt getJSON() og er brukt i de fleste av HTML sidene.

Jeg må innrømme at jeg brukte lang tid på forstå den asynkroniske naturen til Javascript. I en lang stund prøvde jeg å bare lagre dataen fra callbacken direkte i en global variaben uten noe særlig lykke. Problemet er at javascript oppfører seg «multi-trådet» altså at den arbeider på flere funksjoner, eller deler av koden på samme tid, og returnerer når en eller annen del er ferdig. Hvor lang tid dette tar varierer også. Jeg eksperimenterte også med Promiser, men endte opp å gå tilbake til callbacks da jeg fant en akseptabel løsning på det.

Løsningen jeg gjorde var altså at jeg handterer dataen direkte i callback funksjonen som blir en parameter til funksjonen getJSON(). Callbacken selv har dataen som jeg er interessert i som parameter. Derfor kan jeg nå den der ved å for ekempel passere data til en funksjon createList(data) som lager en html liste av alle toalette i dataen eller noe lignende.

Oppgave 3 Toalett-funksjon

Teknisk sett er denne funksjonen implementert som en window.onload funksjon som betyr at jeg laster ned JSON dataen umiddelbart når siden lastes inn. Dette gjelder alle sider utenom «index.html». URL-en til JSON-filen blir tatt som en parameter funksjonen getJSON fra del 2.

Oppgave 4 Oversikt over Toalett

Funksjonen DisplayDefaultData() i «toaletter.js» er i hovedsak ansvarlig for å printe en nummerert liste med toalettnavn, og markører på kartet som samsvarer med dem. Jeg opplevde at å bruke Google-Map API-et var relativt enkelt og problemfritt, og hadde ikke noen særlig utfordring med denne oppgaven.

Oppgave 5 Avansert søk, hurtigsøk
Jeg opplevde at Implementeringen av det avansert søk og hurtigsøk var garantert det mest tidskrevende å sette sammen.

Grunnet kompleksiteten til koden vil jeg skrive enn liten bruksanvisning her. Jeg har kommentert koden i toaletter.js kanske dyptgående, så derfor vil jeg ikke skrive så mye om hvordan koden fungerer her.

HURTIGSØK:
Ved hurtig-søk trykker man ENTER for å søke i feltet. Ved vanlig tekst-input vil koden sammenligne inputtet med navn og adresse til toallettene, og så generere en liste med matcher.

Man kan også gjøre objekt-søk ved å skrive for eksempel: herre:1, kjønn:herre, pris:30 osv. For øyeblikket er det bare mulig å gjøre et objekt-søk i feltet om gangen. For å støtte flere objekt-søk må jeg implementere et mer komplisert regex-objekt. Akkurat nå grupperer regexen teksten til venstre for et kolon tegn (:) og teksten til høgre for dette tegnet. Nøkkelen blir lagret i indeks 0, og verdien blir lagret i indeks 1. Med dette var det mulig for meg å opprette et regex-søkeobjekt som tar indeks 0 og 1 som nøkkel og veri.

AVANSERT SØK:
På kjønn er det mulig å velge både herre, og dame. Derfor er dette inputtet implenterte som «check-boxes» i stedet for «radio-boxes».
Legg merke til at det er mulig å også for eksempel spesifisere at toalett skal IKKE ha rullestol-støtte, eller IKKE våre åpent i dag osv. Jeg mener jeg har implementert dette på en relativt god måte som ikke krevde mye mer kode enn det var fra før av.
Forbedringer:
Så vitt jeg vet fungerer nærmest alt på denne siden som det skal. Det eneste som jeg trur mangler er muligheten til å spesifisere flere objekt søk på samme gang i søkefeltet. For eksempel å skrive både: rullestol:1, og stellerom:1 på samme tid i feltet. Det vil kreve et litt mer komplisert regex oppsett som jeg ikke har laget enda. En mulig løsning kunne vært å spesifiere en regex som grupperer keys som partals grupper(0, 2, 4...) og values som oddetal grupper(1, 3, 5...) ved hjelp av parantes (). Ellers hadde det kanskje også vært greit å kunne velge mellom ukedag, lørdag eller søndag på «Åpen Nå» i det Avanserte søket. For øyeblikket tar den kun klokkesletter for ukedagen i dag.

Oppgave 6 Oversikt over Lekeplasser
Siden viser en enkel oversikt over lekeplasser på et kart, og en liste som med navnene deres. Jeg gjenbrukte et par funksjoner fra tidligere oppgaver.
Det er mulig å velge antall lekeplasser som skal vises i inputt-feltet som finnes til venstre på siden. I forhold til de andre sidene var dette garantert den siden jeg brukte minst tid på. Stylingen kan trenge en del arbeid, spesielt når man ønsker å vise nærmt 100 lekeplasser på en gang, og listen blir veldig lang.

Oppgave 7 Avstand-funksjon.
findDistance() finner avstanden mellom two koordinater gitt fra parameteren.
Mattematikken er basert på Haversine Forumla som antar at avstanden mellom to punkter er en direkte luftlinje. Noe av matematikken er jeg kanskje ikke sikker på, men det vikti. Jeg mener at dette er et godt eksempel på abstraksjon i programmering. Selv om jeg kanskje ikke forstår all matematikken som foregår, så vet jeg hvor og hvordan jeg kan bruke metoden til å få resultatene jeg vil ha.

Oppgave 8 Favoritt-funksjon
Jeg valgte toalett-datasettet for denne oppgaven. Jeg har option/select html-elementer som kan trykkes på av brukeren for å velge et favoritt toalett. Når valget er gjort vil koden kjøre funksjonen fra oppgave 7 som finner nærmeste toalett og returnerer indeksen til dette toalettet i datasettet. Denne indeksen lar oss så lage en liste av nformasjon om toalettet.

Oppgave 9 Min Favoritt Lekeplass
favoritt.html består i hovedsak av 4 deler på en rad:
Valg av toalett,
Informasjon om det valgte toalettet,
Avstand til nærmeste toalett,
og Informasjon om nærmeste toalett.
For brukeren er det mulig å trykke på navnene i listen over toalettet for å velge sin «favoritt». Denne listen består av option/select html-elementer.

Forbedringer:
For øyeblikket er informasjonen som vises om toalettene hentet rett fra de uforandrede verdiene til nøkkelen i datasettet . Dette ville vært bedre å presentere informasjonen på en litt mer mennesklig lesbare måte. For eksempel i stedet for å skrive: Rullestol - «NULL, gir det mer mening å vise RulleStol – nei, eller noe tilsvarande.

Oppgave 10 Vilkårlig Datasett
Datasettet jeg valgte omhandler TV-episoder av programmet «Norge Rundt». Her er det informasjon om episoder fra år 1976-2016, med ulike attributter som: «tema for episoden», «kommune», «kjønn av karakterer», «alder av karakterer» og mye mer. Formatet jeg valgte er JSON.

Det er mulig å spesifisere ulike parametere når en vil hente data fra dette datasettet.
Som et eksempel er dette et utdrag for koden jeg skrev i den tilhørende skript filen - «custom.js»:
var aar = «2016».
var url = «https://hotell.difi.no/api/json/nrk/norge-rundt?aar="+aar+"&kommune=Bergen»
Her er jeg bare interessert i episoder der kommunen er «Bergen», og selve året til episoden er en variabel, noe som betyr at vi kan få ulike datasett dersom vi bare forandrer verdien til år-variabelen.

VIKTIG
Jeg innså ikke før de var for sent å forandre at det står i oppgavekravene at det ikke er tillat å bruke «noen eksterne ressurser» i oppgavene. I «custom.html» (Oppgave 10) har jeg brukt en javascript fil som jeg lastet ned fra: https://canvasjs.com/. Denne filen er kalt (canvasjs.min) og ligger sammen med de andre javascript filene. Det eneste den eksterne ressursen gjør er å lage selve grafikken til kakediagrammet. «custom.html» linker til to javascript filer. All kode i «custom.js» er altså skrivet av meg selv. Dette inkluderer datainnsamlingen/formateringen/logikken og alt annet.
Hvis dette diskvalifserer oppgavedel 10 så laget jeg en alternativ versjon som ligger i HTML-filen «customv2». Denne versjonen bruker ikke noen eksterne ressurser. Forskjellen her er bare at dataen blir bare vist i en liste i stedet for i en graf.
VIKTIG