

HOVEDRAPPORT for “Felles-skapet” - How We Roll

Gruppe 2b

For å kunne begynne oppgaven satt hvert medlem seg ned og leste oppgaveteksten nøye.

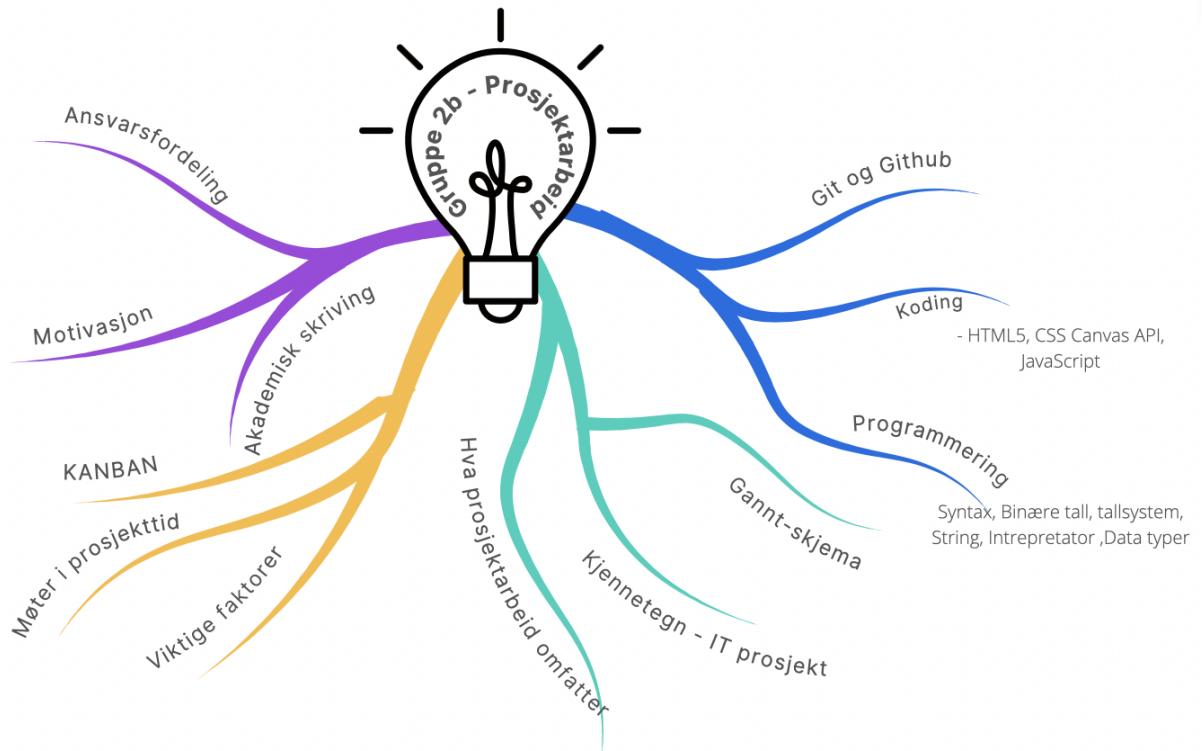
Deretter snakket vi sammen om hvordan vi på best mulig måte kunne fordele arbeid. For å kunne fordele arbeidsoppgaver på en god og oversiktlig måte lagde vi et Github project som tilsvarer et KANBAN-board og delte oppgaven opp. Som gruppe bestemte vi oss for at alle skulle være så mye som mulig med i alle stegene. Derfor har vi valgt å dele opp oppgaver slik at det er minimum to på hver oppgave og at medlemmer med forskjellig kompetanse kan jobbe sammen, for å få mest mulig læringsutbytte.

Gjennom prosessen ble verktøy som; “JavaScript from Beginner to Professional” boken, Mozilla Developer Network og W3 Schools hyppig brukt til å undersøke koding. Det ble brukt til alt fra veldig enkel HTML til det vi syntes var mer avansert/vanskelig. For å kode brukte vi verktøyene Visual Studio Code og IntelliJ. Selvsagt brukte vi også Github, som er der hele prosessen har vært og alle har fått god kontroll på å jobbe sammen (push/pull requests). Til slutt tok vi også i bruk boken Prosjektledelse av Jan Terje Karlsen for mer strukturert prosjektarbeid.

GR-KUNN-FERD

Så langt i dette semesteret har vi lært mye om prosjektarbeid. Vi har lært en del programmering og fått en introduksjon til gode samarbeidsverktøy (Github) av Janis. Samtidig som vi har lært utrolig mye om prosjektarbeid teori av Nils.

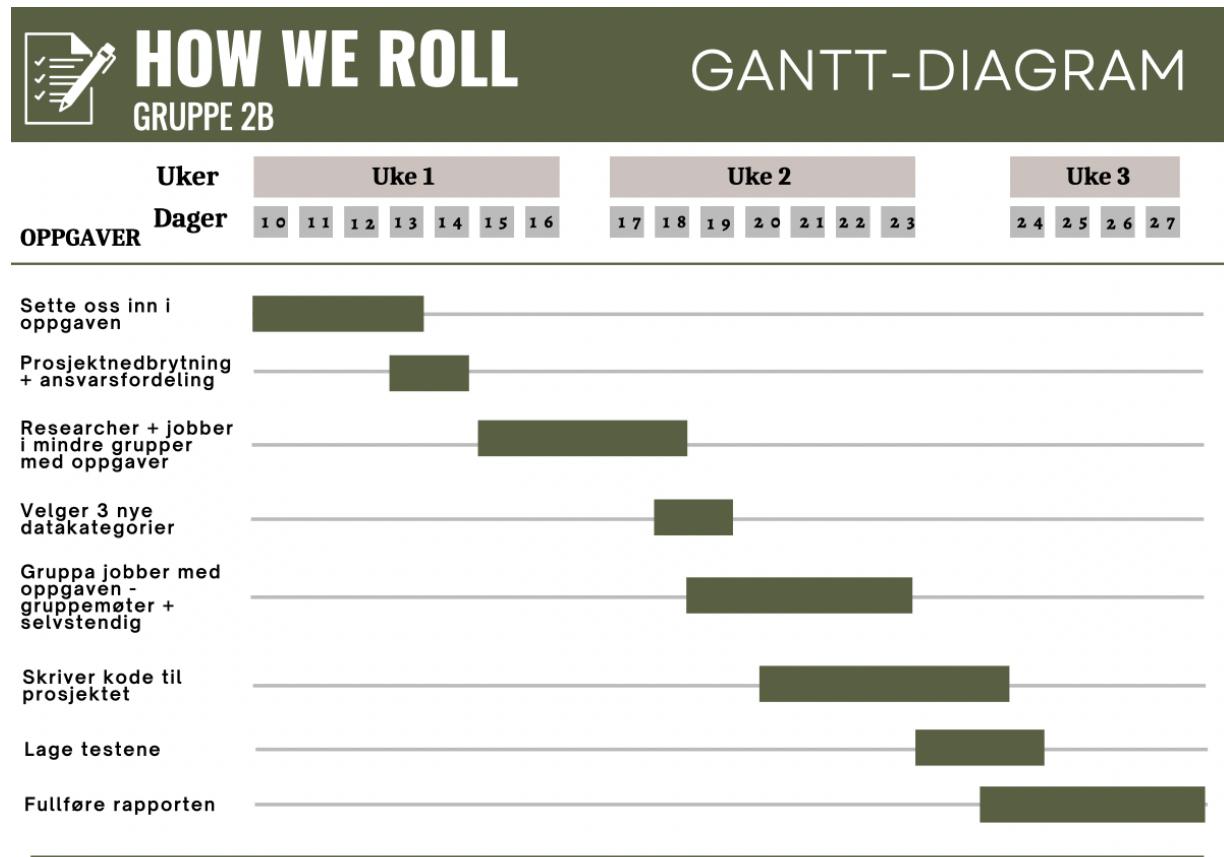
Mindmap av hva vi i gruppe 2b har lært:...



Bildet over er et tankekart med hovedelementene av hva vi har lært så langt. For denne oppgaven er det mye som er relevant. Selvfølgelig vil Git og Github være essensielt da dette er et gruppeprosjekt som krever at vi er mange som jobber sammen på samme kode. Derfor må alle gruppemedlemmer bruke det de har lært om github, spesielt pull og push requests, commit messages og merging for at vi skal kunne jobbe godt sammen. Siden produktet er en webapplikasjon er vi nødt til å ta i bruk det vi har lært om koding og programmering. Som sagt har vi allerede litt kunnskap om HTML 5, CSS, Canvas API og litt JavaScript. Vi tilegnet dette gjennom forelesninger og flagg oppgaven, denne kunnskapen må vi ta i bruk da den er høyest relevant til prosjektet.

Vi må også bruke teori Nils har lært oss for å gjennomføre prosjektet på best mulig måte. Det vi vil bruke er; Kanban-board (det lager vi i github projects), teori om motivasjon (ha klare mål,

god kommunikasjon, en grei arbeidsmengde, gi folk ansvar og jobbe sammen som en gruppe). Teorien om hvordan holde møter (kalle inn til møter, ha en plan for møtet i forveien/agenda, skrive referat og lytte til hverandre). Vi vil også lage et Gantt-diagram. I og med at vi ikke har jobbet med et prosjekt som dette før og vi kun har 16 dager på å fullføre bestemte vi oss for å lage et Gantt-diagram. Dette er for å kunne best mulig planlegge tiden vår og for å skape en struktur innad i gruppen. Gantt-diagrammet vårt fremstiller de planlagte aktivitetene som tykke streker, der lengden på strekene viser varigheten av aktiviteten (Karsen, 2021).



(Vi lagde gantt-diagrammet før fristen ble utsatt, og vi som gruppe bestemte oss for å følge diagrammet så godt vi kunne og ha det som mål å bli ferdig til første frist - 28. Okt)

FS-ANALYSE

Som en team vil vi ha en undersøkelse funksjon hvor hvert spørsmål har to motsatte personlige preferanser og deltakere får muligheten til å gi meningen med å dra et punkt til innen et bestemt linje, slik at vi kan hente dataen fra undersøkelsen og bestemme hvordan vi skal jobbe.

Slik at jeg kan hente denne dataen for å lage et rapport , vil jeg som en deltaker at mine svar lagres i et eksternt server med en url.

Akseptansekkriterier : Vi må ha minst 10 range sliders i rapporten som gir informasjon om alle i gruppen. Deretter må de samles og det blir lettere å se gjennomsnittsverdier.

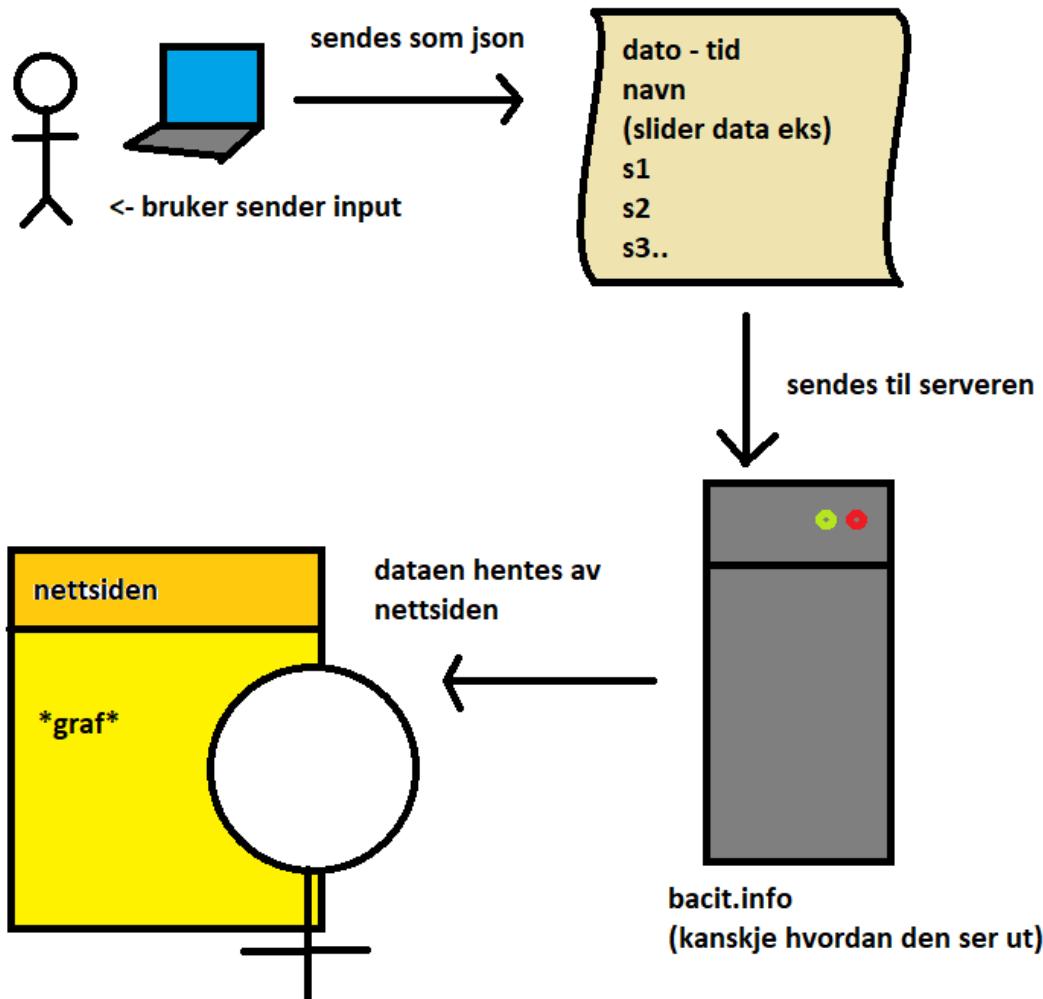
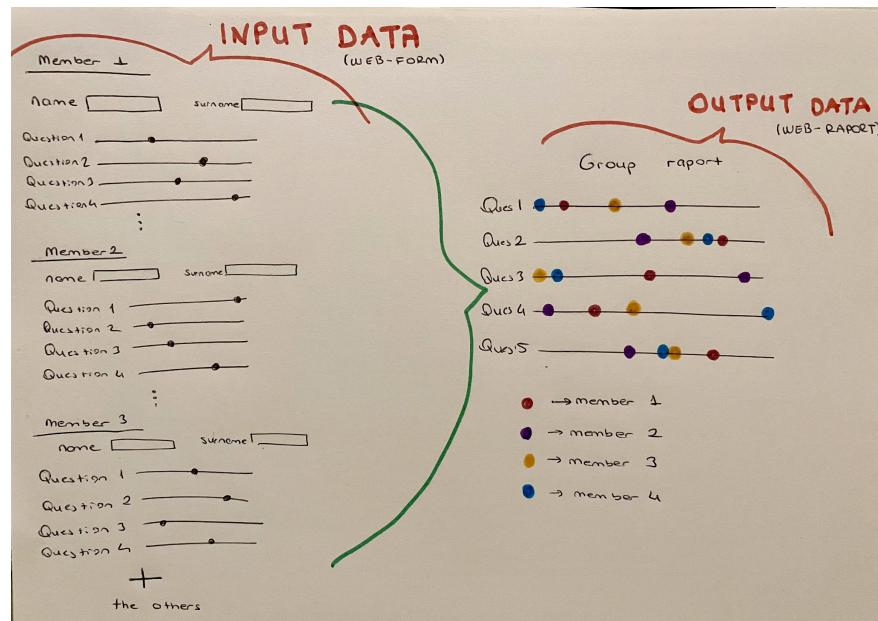
F.eks Mie er gruppe lederen av gruppe 2B og hun ville ha en ekstra møte med medlemmer for å planlegge 114 oppgaven. Hun sjekker how we roll rapporten og ser at mest av medlemmene foretrekker å ha møte på morgen og på discord. Da kan Mie invitere folken til gruppemøte søndag morgen via discord.

Etter å ha jobbet sammen i et par måneder, har vi observert at noen medlemmer, for eksempel, ikke kan fullføre oppgavene sine om morgen. Men noen gjør det. For å overkomme denne utfordringen planla vi gruppemøtene våre til både tidlig morgen og ettermiddagen. Ved å gjøre det kan hvert medlem gjøre sitt beste, enten om morgen eller senere på dagen. I tillegg bruker vi Discord for å varsle alle medlemmer om kommende møter. Dessuten for å holde dem engasjert på nettet. Vi har fått mye kunnskap om vaner og uvaner til våre medlemmer.

How We Roll vil hjelpe oss med å skaffe mye informasjon om medlemmene våre. Alle spørreundersøkelser vil bli lagt til kunnskapen vi tidligere har fått om våre medlemmer gjennom en samarbeidsprosess. Dette vil hjelpe oss i å lære mer om vaner til medlemmer og lære hvordan vi kan fikse uvaner. Som et resultat vil vi kunne overkomme utfordringer som gjør at vi misforstår, mangler kommunikasjon og ikke klarer å fullføre oppgavene i tide.

FS-SYSTEM

For å oppnå oppgaven via skissen sett over, må vi:



FS-SYSTEM-STEG

Skissen fra FS-SYSTEM er en abstrakt skisse til hvordan systemet vårt skal fungere. Med systemet prater vi om å omgjøre brukerens input til brukbar output. Istedentfor at vi går rett på kodingen, blir det mer oversiktlig hvis vi bryter oppgaven ned til mer abstrakte modeller.

Hvordan input dataen skal sendes inn og hvordan output dataen blir hentet vil bli slik:

- Input
 - Brukerens fornavn og etternavn i to forskjellige tekst input bokser.
 - Denne dataen må fylles ut før brukeren kan sende inn sin besvarelse, dette er for å unngå systemfeil.
 - Slider dataen, brukerens meninger om spørsmålene vi stiller hverandre.
 - Denne dataen blir lagret som talldata.
 - På grunn av at det er flere sliders kunne dette blitt lagret som array, men det kompliserer oppgaven mer og er ikke nødvendig fordi antall sliders er statisk.
 - Dato når dataene blir sendt inn.
 - Dette vil egentlig ikke brukeren fylle ut, men heller gjøres automatisk av nettsiden.
 - En individuell id.
 - Dette kan løses med et veldig stort tilfeldig tall. Tallet kan dobbelsjekkes med de eksisterende tallene i serveren slik at de ikke 'klasher'.

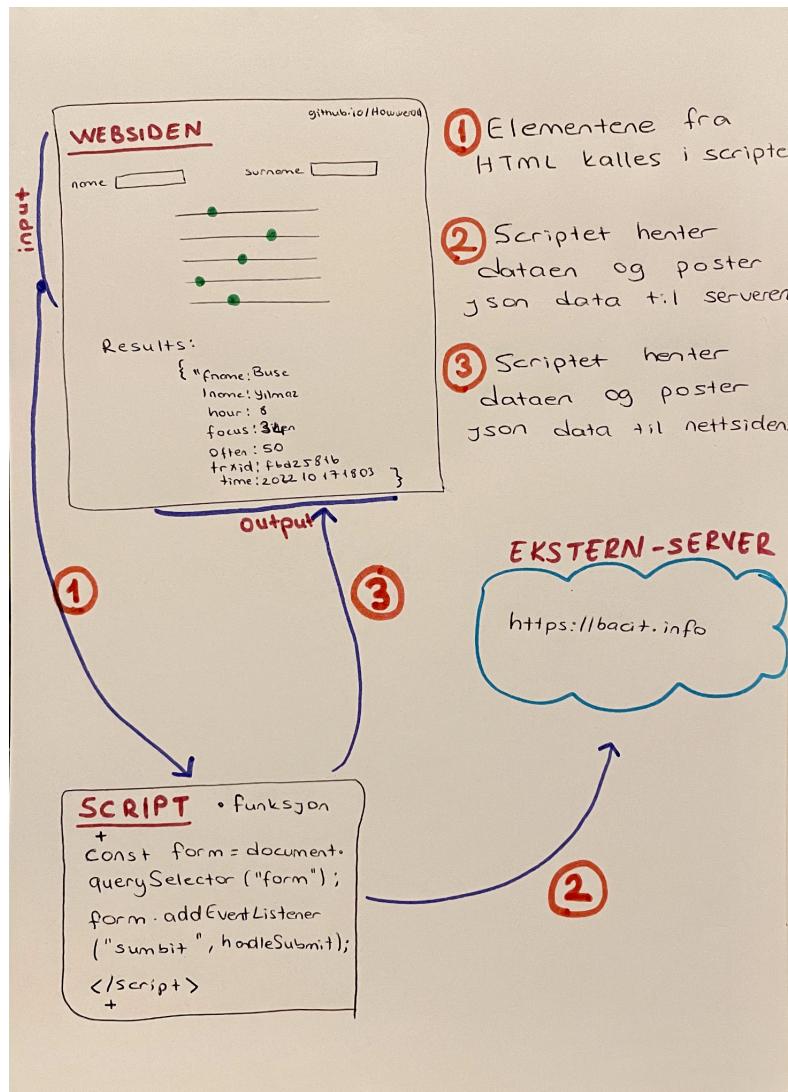
Denne input dataen vil bli sendt fra nettsiden til EKSTERN-SERVER. Neste steget nå er å kunne hente og lese denne dataen.

- Output - All dataen fra EKSTERN-SERVER vil bli hentet inn når siden lastes ned og den vil bli lagret slik:
 - **Class**, data satt inn i ei struktur. I klassen må vi ha:
 - Navn
 - Korresponderende id
 - Hver individuell tall verdi til hver slider, som sagt kunne dette være en array men det blir mer komplisert enn nødvendig

I det fjerde siste punktet var “class” nevnt og forklart som strukturert data. Ved å bruke klasser, som blir dannet til objekter, vil gjøre koden mye renere og lese og mye lettere for å samarbeide i koden.

Hvordan informasjon samles inn om prosjektmedlemmene og resultatene presenteres

Som står i GR-KUNN-FERD har vi lært allerede å bruke systemer f.eks. html, css, kanban og github. Først laget vi website template. Og deretter laget vi 12 forslag til data kategorier i websiden fanen, operert folkens egne egenskaper med WEB-FORM. Det betyr data fra brukerne er samlet for rapporten. Dataene som vi lagde er input. Etter er alle ferdig med å legge inn data, vi laget en forståelig rapport viser at vaner av gruppe medarbeider. Så er dataene fra WEB-FORM behandlet som output i WEB-RAPPORT.

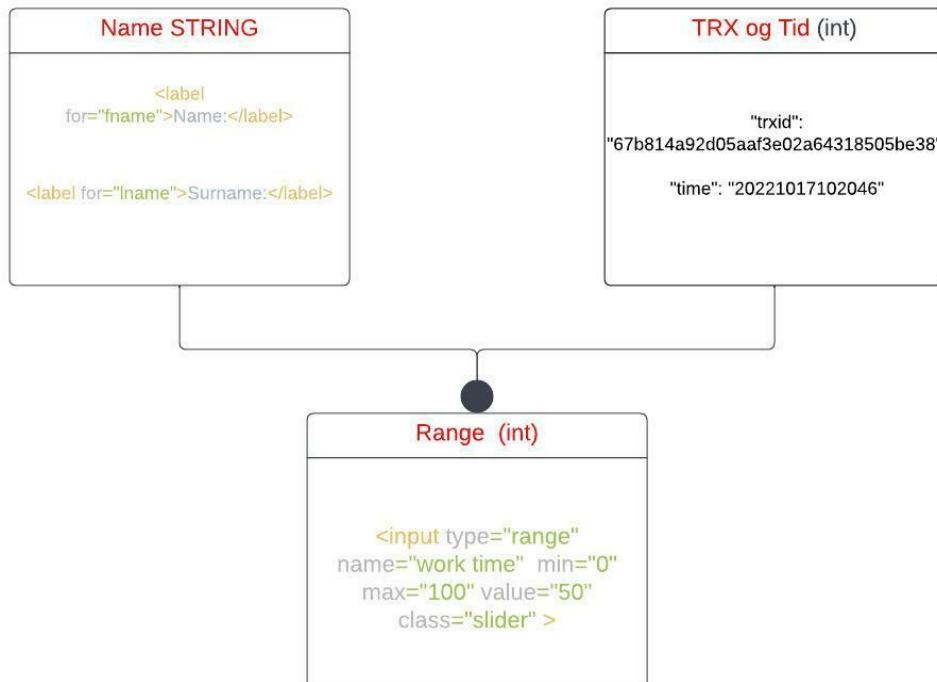


Html attributene fikk navn i forhold til funksjonen til elementen. Disse navnene ble valgt slik at andre som ser på koden kan forstå hva elementen inneholder. Siden disse elementene har fått navn som klasser, fikk vi muligheten til å kalle disse fra css filen og samtidig script elementet som inneholder funksjonelle javascript kode.

I css kalte vi “slider”, “slidercontainer”, “button” og samtidig andre globale elementene (sånn som body) for å gi stil til FS-SYSTEM. I script kalte vi “form” og “submit” elementene for å kunne hente dataene fra html. “Form” elementen inneholder dataen og “submit” elementen kobler knappen til script funksjonen. På denne måten har vi gjort prosessen fra html input til json output.

Datamodellen

Når det gjelder datamodellen, har vi valgt å bruke to string elementer som er navn og etternavn, tolv nummer elementer som kommer fra html range elementene, og samtidig transaksjons-id og tid-stempel. Disse lagres i en JSON object literal.



FS-TESTER

Ved å se på FS-SYSTEM-STEG og funksjonene som hører med stegene vil det være nødvendig, i henhold til kravspesifikasjonen å teste dette:

Her beskriver vi funksjonene som gir oss ønsket output:

1. Brukerscenario: lage en fungerende range slider

- Verktøy: HTML 5, CSS
- Egnede funksjoner: <input type="range"> (html element, ikke funksjon)
- Testfunksjon som påkaller funksjon: .slider (kalles i css men ellers er elementet visst i nettleseren.)
- Utfør testfunksjon:

```
<td colspan="3"> <input type="range" name="worktime" id="worktime" min="0" max="100" value="50" class="slider" ></td>  
(tabell element er visst i nettleseren med verdier sånn som navn, min, max osv.)
```

2. Brukerscenario: Etter å ha fylt inn min informasjon i range slidersene trykker jeg på submit button og får resultatene (hvordan de lagers på ekstern server)

- Verktøy: HTML, JavaScript
- Egnede funksjoner: function postData(event) {}
- Testfunksjon som påkaller funksjon: function handleSubmit{}
(postData kalles i handleSubmit funksjonen.)
- Utfør testfunksjon:

```
async function postData(url ='https://bacit.info/', data = {}) {  
    const response = await fetch(url, {  
        method: 'POST',  
        mode: 'cors',  
        cache: 'no-cache',  
        credentials: 'same-origin',  
        'Content-Type': 'application/json'},  
        redirect: 'follow',  
        referrerPolicy: 'no-referrer',  
        body: JSON.stringify(data)  
    "Content-Type"});
```

```

        return response.json();
    }

    (Metoden som kalles for å poste data i bacit.info)
    postData('https://bacit.info/', formJSON)

```

3. Brukerscenario: Deretter må vi kunne hente resultatene i serveren bacit.info

- Verkøy: Html, JavaScript
- Egnede funksjoner: function handleSubmit(event)
- Testfunksjon som påkaller funksjon: data => {}

(denne metoden lar oss hente dataen med en Url som har trxid og tid)
- Utfør testfunksjon: .then(data => {

let trxid = JSON.stringify(data.trxid, null, 2).replace(/\//g,

"");

let time = JSON.stringify(data.time, null, 2).replace(/\//g, "");
 });

4. Brukerscenario: Tegne en rektangel i nettleserens vindu

- Verktøy: html,css, javascript, Canvas API
- Egnede funksjoner: function tegnKvadrat() {}
- Testfunksjon som påkaller funksjon: function submitButton() {}
- Utfør testfunksjon: <button onclick="submitButton"> </button>
- function tegnKvadrat() {"ctx.beginPath(); ctx.rect(x1,y1, x2, y2);

ctx.stroke();
 "}

5. Brukerscenario: Tegne linjer og skrive tekst inni rektangelen

- Verktøy: html,css, javascript, Canvas API
- Egnede funksjoner: function tegnLinjeOgSkrivTekst() {}
- Testfunksjon som påkaller funksjon: function submitButton2() {}
- Utfør testfunksjon: <button onclick="submitButton2"> </button>
- function tegnLinjeOgSkrivTekst() {Linje: ctx.beginPath(); ctx.moveTo(x,

y); ctx.lineTo(x, y); ctx.fill(); }

6. Brukerscenario: Tegne sirkler i bestemte farger og størrelser

- Verktøy:html,css, javascript, Canvas API
- Egnede funksjoner:

```
function tegnSirkler() {}
```
- Testfunksjon som påkaller funksjon:

```
function submitButton3() {}
```
- Utfør testfunksjon:

```
<button onclick="submitButton3"> </button>
```
- ```
function tegnSirkler() {Sirkel: ctx.arc(x, y, 10, 0, 2 * Math.PI,
false); ctx.fill();
}
```

## GITHUB-SAMARBEID

Som sagt i GRUNN-KUN-FERD begynte vi prosessen med å opprette en ny repository for How We Roll prosjektet. Deretter hadde vi et gruppemøte for å forstå oppgaveteksten og fordele oppgaver. På dette møtet lagde vi en Github Project og brukte det som kanban boardet vårt for prosjektet. I tillegg til dette lagde vi et gantt diagram, men fristen ble utsatt så vi endte opp med å bruke lengre tid, og dermed ikke følge kanban boardet. Noe som vi har reflektert over nå i senere tid og syntes var dumt, derfor vil vi i fremtiden bruke det mer aktivt for å kunne planlegge tide bedre.

Samarbeidet i Github:

Vi har på gruppemøtene våre avtalt at vi ikke skal gjøre noen endringer i main/master branch til repository-et. Derimot, eventuelle endringer som må brukes i prosjektet til HTML-, CSS- eller Javascript-koden eller i hele prosjektet må publiseres i en ny branch. I tillegg, når det gjelder å slå sammen branches til main/master branch, kan ingen av medlemmene trykke merge uten først å fortelle de andre medlemmene om merging process og forventet endringer.

Som sagt brukte vi projects i Github for å fordele oppgaver, noe vi syntes var nyttig og ble brukt aktivt. Vi brukte kanban-board for å reflektere over prosjektets krav. Kanban hjalp oss med å organisere oppgavene for dette prosjektet ved først å plassere elementene i backlog-kolonnen.

Deretter, når vi har kommet i gang med en oppgave kan den flyttes videre til kolonnen 'in

progress'. Dette er hjelpsomt for å selv holde styr på oppgaver, men er også kjekt for de andre gruppemedlemmene. Da kan man se hva som mangler, hvor langt alle har kommet i arbeidet og om noen trenger hjelp på en oppgave. Når en av medlemmene har fullført sin del, blir den manuelt flyttet til 'Review-kolonnen', hvor vi diskuterer enten vi skal godkjenne/redigere disse delene, eller om det trengs mer arbeid. Til slutt vil alle krav til prosjektet som er ferdige bli dratt til 'Done-kolonnen', hvor alle vil være klar over at den delen har blitt gjennomført vellykket. Samarbeidsprosessen vår i gruppe 2b har vært over forventning og vi har fullført prosjektet selv om vi har møtt et par hindringer i veien.

Disse hindringene har vært:

- *Hindringer innad i gruppen:*

I løpet av tiden vi har hatt How We Roll oppgaven har vi som gruppe hatt noen problemer innad. Det har tatt tid og krefter at vi har måtte dratt til rådgiver, snakke og prøve å løse problemer på gruppemøter, og mangel på oppmøte i blant. Derfor har dette satt press på oss som gruppe og vi har brukt tid og krefter på dette, som egentlig kunne blitt brukt på oppgaven. Men dette er noe som har vært nødvendig å ta tak i. Vi mener at som gruppe har vi kommet sterkere ut av det og lært mye om samarbeid i løpet av prosessen.

- *Arbeidsmengde og håndtering av tid*

Samtidig som How We Roll oppgaven har vi hatt ukesoppgaver, deliverables, semesteroppgave osv. Det er store arbeidsmengder som skal leveres rundt samme tidspunkt og derfor har det vært krevende med tidshåndtering og prioriteringer av oppgaver. Med en gang vi fikk How We Roll oppgaven begynte vi å jobbe intensivt. Dette er vi utrolig takknemlig for nå fordi det har gjort det lettere for oss å jobbe videre med How We Roll, selv når vi fikk andre oppgaver. Derfor, selv om det har vært krevende med store arbeidsmengder til tider syns vi at How We Roll har blitt prioritert på en god måte og er fornøyde med arbeidet til slutt.

- *Forståelse av all oppgavetekst*

Oppgaveteksten syns vi som gruppe var litt trøblete å forstå oss på. Vi brukte en god del tid på dette og trodde vi hadde forstått alt til slutt. Vi spurte Janis om ting og LA. Derimot merket vi mot slutten at vi hadde misforstått FS-TESTER og kun skrevet i naturlig språk hva vi ville testet (ikke hvordan). Dette var en nedtur og hindring for oss på veien. Men vi spurte mer og fant ut av det til slutt.

- *At alle skal commite/skrive kode når vi er så mange på gruppa*

Vi er som sagt mange på gruppen og det er vanskelig om alle skal comitte litt kode hver. Derfor er det noen gruppemedlemmer som har veldig få commits og andre som har mange. Derfor er det viktig at vi sier ifra om at alle i gruppa har bidratt, men vi har valgt **å kode 2 og 2 sammen**, siden ingen av oss er spesielt gode eller selvsikre på koding. Vi har sittet sammen og skrevet i par på møter, derfor har det endt opp med at det er de samme som commita mest.

Alt i alt er vi fornøyde med oppgaven og samarbeidet vårt. Vi har brukt github godt og lært masse om både teori, koding, github og hverandre som gruppemedlemmer.

#### Tekst Kilder og Kode kilder

Othneildrew, u.å, Best ReadMe Template, ukjent, Source code:

<https://github.com/othneildrew/Best-README-Template>

W3schools, u.å, How TO - Range Sliders, ukjent, source code:

[https://www.w3schools.com/howto/howto\\_js\\_rangeslider.asp](https://www.w3schools.com/howto/howto_js_rangeslider.asp)

Mozilla, u.å, <center>: The Centered Text element, ukjent, source code:

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/center>

Codingartistweb, u.å, Custom Range Slider, ukjent, source code:

<https://codingartistweb.com/2021/07/custom-range-slider-html-css-javascript/>

W3schools, u.å, Json objects literal, ukjent, source code:

[https://www.w3schools.com/js/js\\_json\\_objects.asp](https://www.w3schools.com/js/js_json_objects.asp)

SheCodes, u.å, Css Gradient Generator, ukjent, source code:

<https://generators.shecodes.io/css-gradient-generator>

Ålvaro, u.å, Untitled, ukjent, source code:

<https://codepen.io/alvarotrigo/pen/mdBbEPj>

Gruppe repo eksempler

<https://github.com/theodorUIA/howweroll>

<https://github.com/stinawest/how-we-roll>