

Intervjuoppgave – Utviklere  
til [mattilsynet.no](https://mattilsynet.no)

# Hvordan er oppgaven blitt løst

- Laget en Node.js applikasjon som serverer brukere et «kontROLSenter»
- KontROLSenteret er i praksis en nettside med knapper som sender web forespørsler til mitt egne api eller direkte til docker container sitt api gitt tilgjengelig med oppgaven
- Valgte å bruke Node.js siden den inneholder noen semi-moderne rammeverk for web utvikling som «express». I tillegg er jeg godt kjent med teknologien og kunne bygge appen effektivt med tanke på dårlig tid.
- Brukte 4 timer på oppgaven, store mengder tid gikk til å friske opp minnet for diverse web utfordringer som «cross origin» og lignende

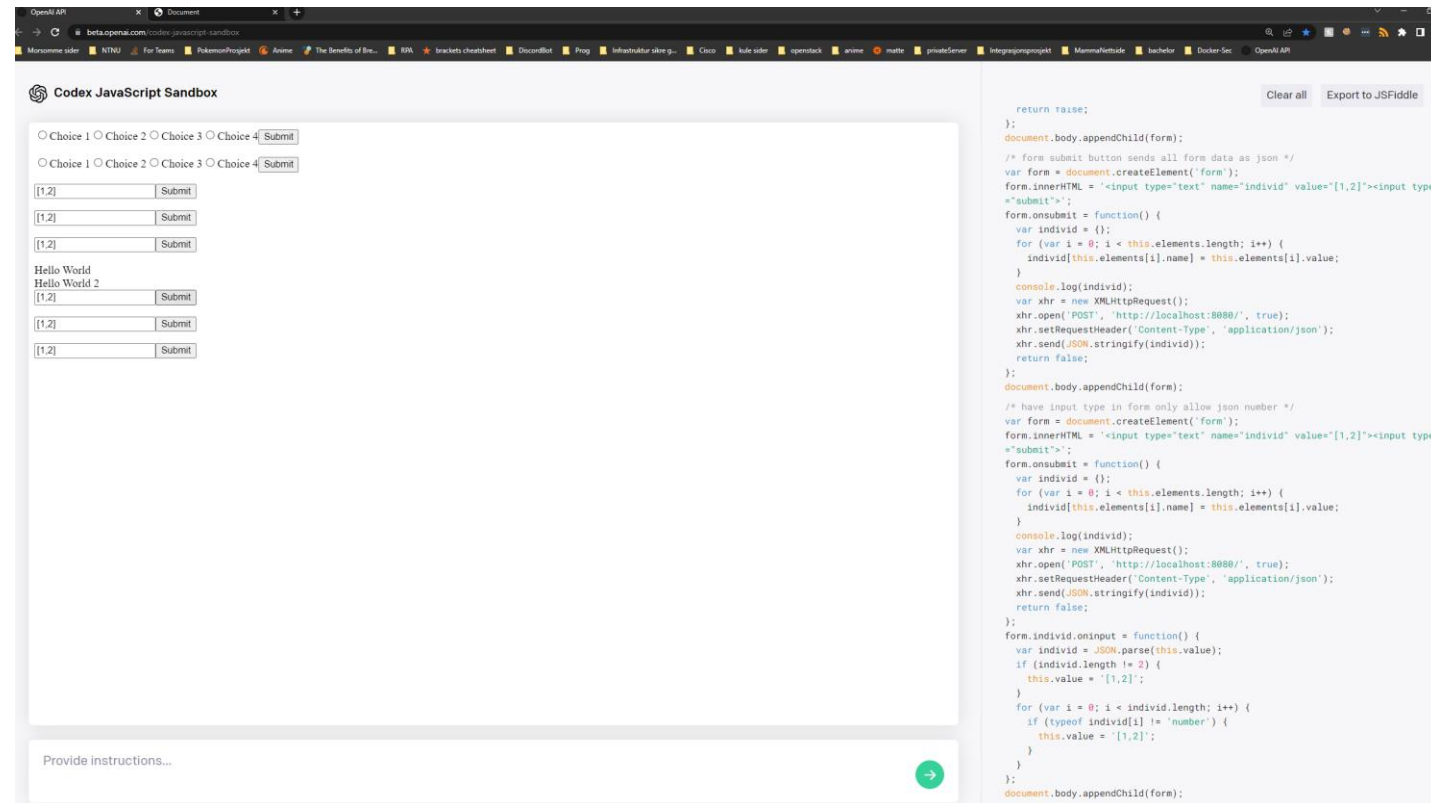
# OpenAi

- Brukte OpenAI underveis i oppgaven for å generere boilerplate kode.

For eks:

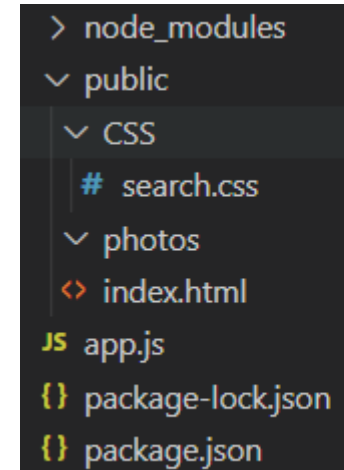
1. Html

2. Web forespørsler



# Installasjon

- Etter installasjon burde folder inneholde følgende:
- Åpne folder med terminal
- “npm install” – commando for å installere diverse node\_modules
- Node app – starter filen app.js med nodde



A screenshot of a file explorer interface with a dark theme. The tree view shows the following structure: a 'node\_modules' folder (expanded), a 'public' folder (expanded), a 'CSS' folder (expanded) containing 'search.css', a 'photos' folder (expanded), and 'index.html'. In the main pane, the following files are listed: 'app.js' (with a yellow 'JS' icon), 'package-lock.json' (with a yellow '{ }' icon), and 'package.json' (with a yellow '{ }' icon).

```
> node_modules
├─ public
│  └─ CSS
│     └─ # search.css
│  └─ photos
│  └─ index.html
└─ app.js
   package-lock.json
   package.json
```

# Rask tilbakemelding

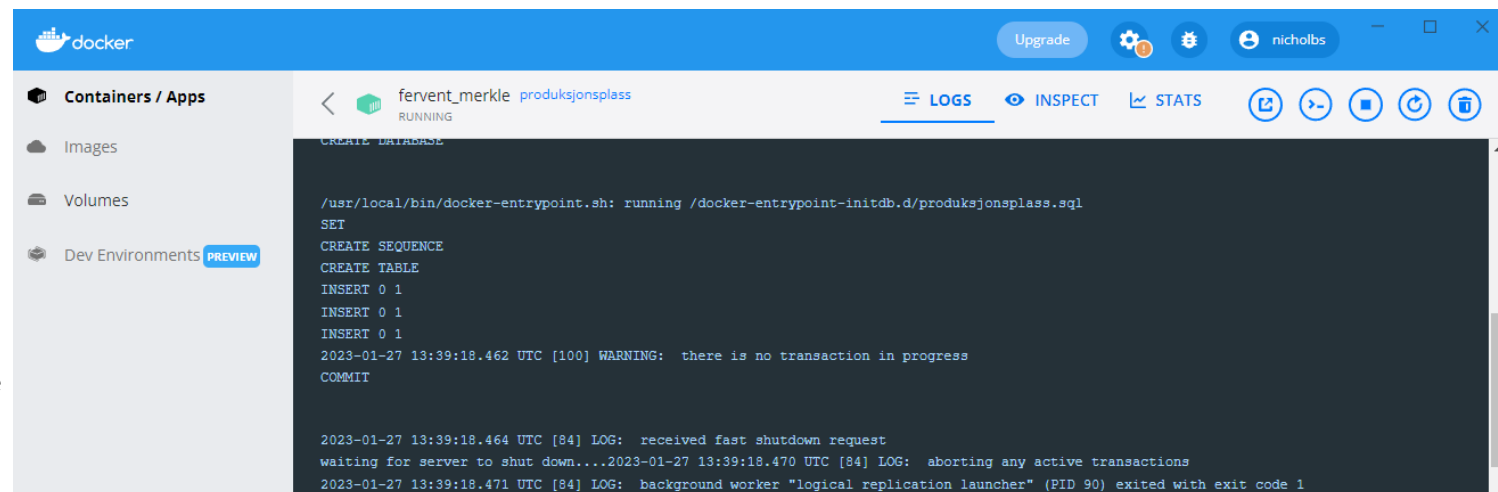
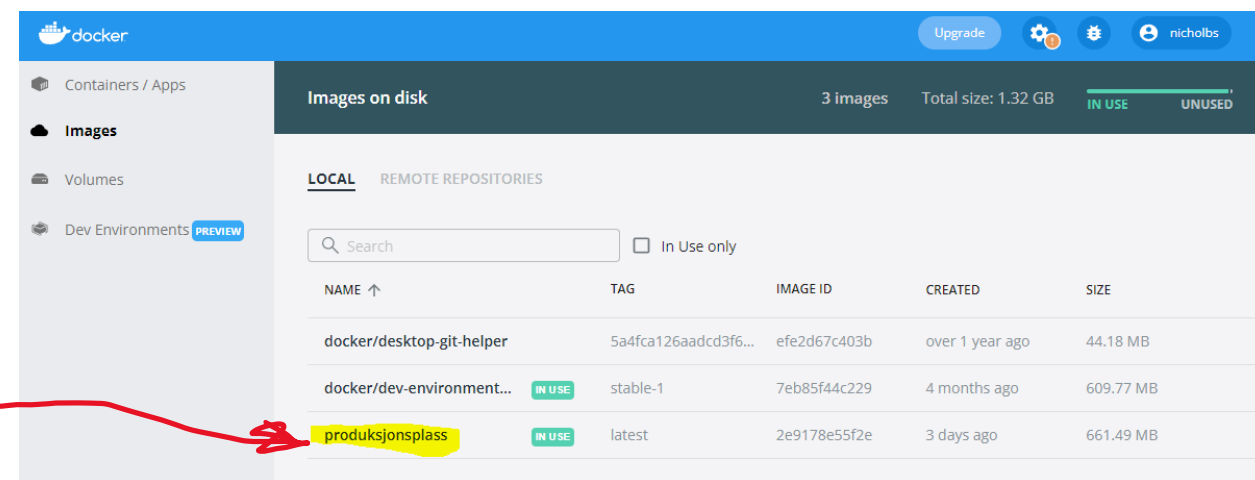
Start kommandoer:

```
unzip produksjonsplass-image.zip
cd produksjonsplass-image
docker build -t produksjonsplass .
docker create --name produksjonsplass-api -p5000:8080
produksjonsplass:latest
```

```
docker start produksjonsplass-api
```

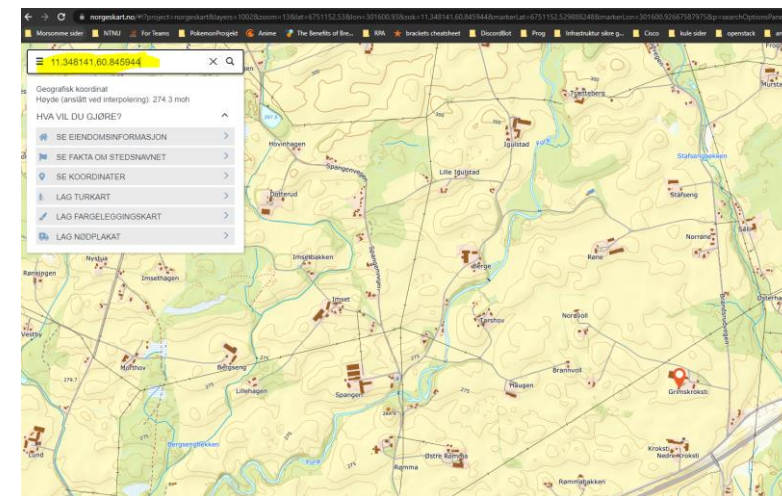
```
#Kommando for å liste ut log fra api'et
docker exec -it produksjonsplass-api tail -f /tmp/api.out
/tmp/api.err
```

Når jeg fulgte kommandoene ovenfor fikk jeg imaget som heter «produksjonsplass» ikke «produksjonsplass-api». Dermed måtte jeg endre På kommandoer for å bygge og kjøre image. På andre siden Kanskje jeg skrev noe feil, for kommandoene ser nokså riktig ut med tanke På dockerfilen gitt med oppgaven.

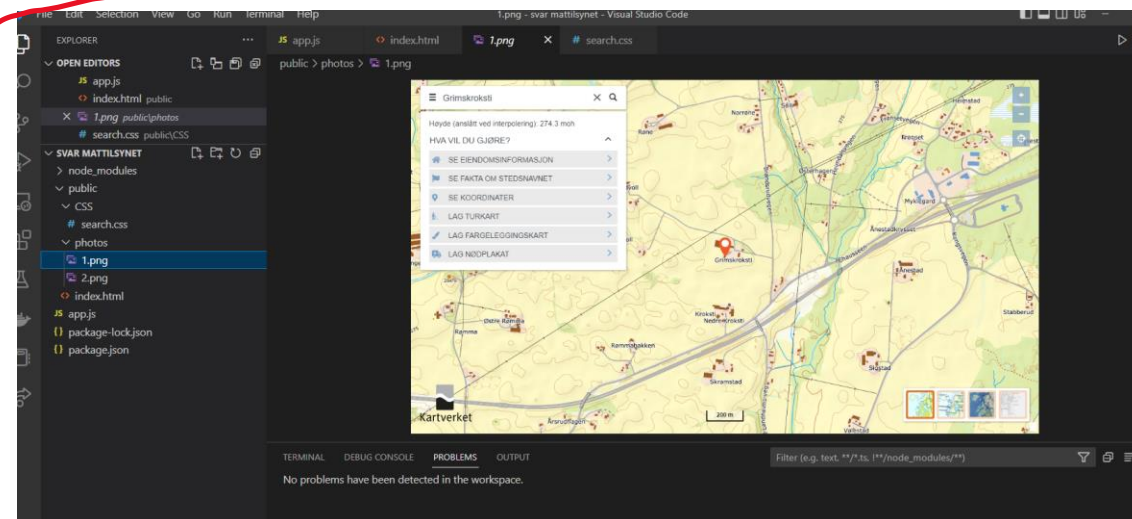
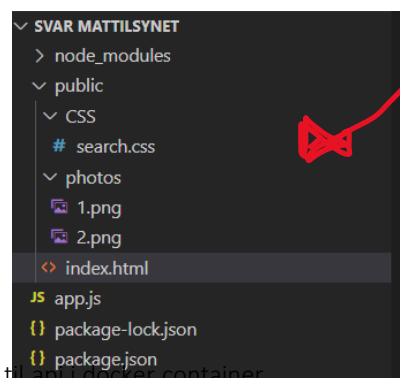


# Oppgave 1

produksjonsplassid	kommunenummer	gaardsnummer	bruksnummer	byggningsnummer	koordinater	Utdrag fra norgeskart.no
1000000	0001	1	1	-1	Mangler koordinater	Beklager, koordinatene var ikke tilgjengelig til å hente et bilde fra!
1000001	0001	3	1	-1	11.348141,60.845944	
1000002	0001	2	1	-1	11.631629,60.31758	



-----forste oppgave-----  
 List ut produksjonsplassene fra endepunktet <http://localhost:5000/produksjonsplass>



1. Bruker trykker på list ut «...produksjonsplass» som sender web forespørsel til api i docker container
2. Lager en «table» basert på svar fra container, i tillegg sender jeg en ny web forespørsel som åpner «norgeskart» ved Bruk av «Google» sin node js modul «puppeteer» og tar skjermbilde av koordindnater hentet ut fra container api.
3. Bildet blir lagret i «photos» folder og html raden mappes til bildet.
4. I praksis blir alle prodduksjonsplassene fra enddepunktet localhost:5000/produksjonsplass listet ut
5. NB, kan være en må refreshe en gang etter å ha trykket på «list ut» knappen for å vise bildene hentet fra norgeskart  
Ettersom jeg ikke har lagt på noe reload/loading logikk for når bildet er mottatt fra puppeteer nettleser instansen som opprettes.

# Oppgave 2

-----andre oppgave-----

Submit

\* velg dyrets ID:

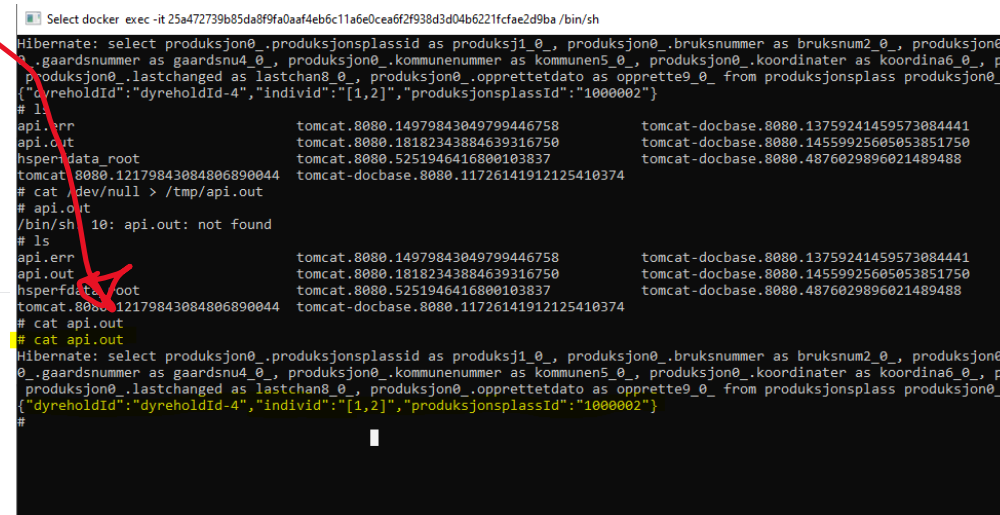
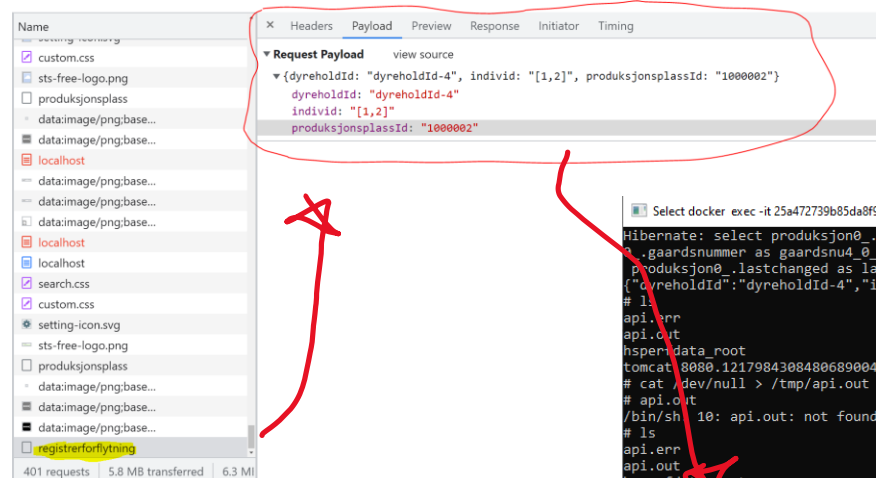
☒ dyreholdId-1 ☐ dyreholdId-2 ☐ dyreholdId-3 ☐ dyreholdId-4

\* Skriv individnummer i format [number, number]:

[1,2]

\* velg produksjonsplassId:

☒ 1000000 ☐ 1000001 ☐ 1000002



- 1.Henter produksjonsplassId fra samme web forespørsel som i oppgave 1.
- 2.Lager en dynamisk form basert på input fra web forespørselen
- 3.Når bruker trykker «submit» senders ny web forespørsel til endepunktet «registrerforflytning»
- 4.Kan deretter verifisere at forespørsel nådde api i docker container ved å åpne terminal til container og lese utskrift fra «api.out» filen

# Hva kunne blitt gjort bedre?

- Mer moderne rammeverk for kode
- Bootstrap eller lignende CSS rammeverk for rask, men også effektiv utseende
- Mangler totalt Sikkerhetsmekanismer. Input fra bruker usikker, nettforespørsler er usikre osv
- Kommentering – pleier å følge doxygen standard, ikke tid til å gjøre nå. Men forklarer gjerne kode der ønskelig
- Moduler – løsningene kunne blitt laget som gjenbrukbare komponenter