# Planleggings dokument fagprøve Victor

## Målet med oppgaven

* Lage underholdnings web-applikasjon for innad bruk i firma
  + Quiz
* Oppnå krav som står beskrevet i oppgaven

## Kravspesifikasjon

* Administrator skal kunne lese inn quizer via fil
  + Admin brukere blir opprettet manuelt i databasen
* Skal ha kapasitet til flere quizer
  + Overordnet quiz navn, f.eks. Quizkommunens fredagsQuiz 12
    - Kategorier, Spørsmål, svar
* Lag til quiz blir opprettet muntlig, 1 maskin (telefon, pc) blir brukt til å gi svar under valgt Gruppenavn som blir skrevet inn på web
* Spørsmål blir stilt i tilfeldig rekkefølge
* Admin skal kunne se resultater til hver enkelt quiz og hvilke team som har deltatt på quizen

## Fremgangsmåte

Listet nedover i rekkefølgen jeg anser som best mulig for utvikling av prosjektet.

* Oppsett
  + Versjonskontroll
  + Hosting av web & database-løsning
* Videre oppsett/ forberedelse før programmering
  + Opprette enkelt design for webside
  + Opprette database-skjema
* Begynne på websiden
  + Mer generelle ting
    - Oppsett for SQL kobling
    - Oppsett for behandling av session
  + Admingrensesnitt (front & backend)
    - Loginn for admin
    - Import av nye quizer gjennom JSON fil
    - Lister alle quizer
      * Om quiz er utvidet, lister alle lag som har besvart quiz
  + Brukergrensesnitt (front & backend)
    - Liste over Alle quizer
      * Quiz blir trykket på, lagnavn blir valgt og quiz starter
      * Kategorier til quiz blir gjort tilfeldig, slik at rekkefølgen ikke er lik
      * Spørsmål kommer løpende etter hverandre når du har gitt svaret ditt.
      * Fullført skjerm, for når team har svart på alle spørsmål

## Potensielle avgrensninger

Avgrensningene som jeg ser, er at det ikke blir lett gjort å lage quizen mer i nåtid.

Oppgaven ber ikke om at quizen skal skje for alle i sanntid, men best mulig sett burde funksjonaliteten vert slik.

### Potensielle forbedringer.

* "loginn", funskjonalitet hvor quizer kan låses ned til at kun de med riktig passord kan gjennomføre quizen
* Admin kan i spesielt grensesnitt starte et samlet spill
  + Oppretter kode som kan skrives inn når man velger lagnavn
  + Admin starter en Nedtelling når det samlede spillet starter.
  + Når nedtelling er ferdig hentes svardata fra databasen som henger sammen med Spillkoden, og viser en score tabell med plasseringer

Ikke sannsynlig at forbedringer under blir gjort

* Søkefunksjon over Overordnet navn for quizer
* Websiden husker brukers svar om de bare har fullført halve quizen.
* Blir da også nødvendighet for brukere som kan finnes frem igjen, innlogging/ oppretting av brukere/ langvarige quizteam
  + Bruker kan velge mellom å besvare quizen som gjest, skriver inn lagnavn og får besvare quizer, men om de ikke blir ferdig med quizen blir ikke svarene husket
  + Om bruker oppretter loginn Kan de se tidligere fullførte quizer sitt resultat og quiz progress blir lagret etter hvert svar

## Skisser av løsningen

Visuell skisse over forslag til databasestruktur før eventuelle forbedringer og utvidelser til oppgaven.

Et bilde som inneholder tekst, skjermbilde, Font, diagram

KI-generert innhold kan være feil.

JSON struktur for fil admin leser inn ligger i fil exampleData.json

Et bilde som inneholder tekst, skjermbilde, programvare, Multimedieprogramvare

KI-generert innhold kan være feil.  
Oppsett for hosting av oppgaven, om det skal bli byttet til hostes lokalt kan serveren for web hoste både SQL db of websiden, og websiden kan åpnes til det lokale internettet.  
Et bilde som inneholder tekst, skjermbilde, datamaskin, design

KI-generert innhold kan være feil.

## Antatt tidsforbruk

Et røft/generøst tidsoverslag for oppgaver som skal gjøres

* Planlegging, 1d
* Arbeid, 5d
  + Forberedelse før programmering 1d
    - Sette opp hosting av Websiden, 1t
    - Postgres database, 4t
    - Planlegge og starte utvikling av utseende til websiden, 3t
      * Quiz spørsmål elementer
      * Admingrensesnitt og innlogging
      * Startside for quiztaker
      * Resultat-panel
  + admingrensesnitt 1d
    - Backend
    - SQL for å hente
      * Quiznavn og resultater til quiz
    - Frontend
    - Loginn
    - Fremvisning av data
  + brukergrensesnitt 2d
    - Backend
    - SQL for å hente:
      * quizer
        + alle child elementer, kategorier, spørsmål/svar
    - SQL for å gi svar på spørsmålene
  + Frontend
    - Listevisning over alle quizer
      * Fremvisning av Kategorier og spørsmål
      * Visuelle elementer når quiz er fullført
  + Tvist, 1d
    - Tid som gjennstår til å bruke på tvisten som kommer, utelat jeg fullfører andre oppgaver på mindre enn forventet tidsbruk.
* Fremføring siste dag, 1d
  + Fremføring av det ferdigstilte prosjektet og forklaringer over hva som skulle/ eller kunne ha blitt gjort bedre, e.v.t diverse forbedringer og avvik.

## Kostnadsoverslag

### Arbeidskostnad

* 1250kr/t, 40t arbeid

### Potensielle kostnader for utvidet hosting

Til utvikling brukes gratis-planer hos Vercel og Neon Gjennom Vercel.

Men om denne web applikasjonen trengs å utvides til et større publikum må det byttes til en plan med mer kapasitet.

* Vercel, Pro. Ca 200kr/mnd
* Neon, Launch. Ca 200kr/mnd

## Programvare, utstyr, ressurser

### Programvare

* GitHub, Versjonskontroll
* Vercel, webtjeneste for Hosting av Next.JS applikasjon.
  + Henter kode fra GitHub repository
* Neon, serverless postgres DataBase. Enkelt opprettet og knyttet til prosjekt, opprettet gjennom Vercel sin platform.
* Rammeverket Next.js.
* @neondatabase/serverless, node module for oppkobling mot Neon sin postgress SQL.
* Bcrypt, node module for hashing.
  + Om systemet skal utvidets til at permanente brukere skal kunne bli opprettet
* Iron-session, node module for Session funksjonalitet for midlertidig lagring av sensitiv informasjon.