Dag 1 – Planlegging

* Skrevet planlegging og gjort steg andre steg som har blitt nevnt og var sammenhengende
* Satt opp GitHub repository,
* Satt opp starten til et Next.JS prosjekt med startkommando
* Fikk repoet hostet hos Vercel, bruker koden som ligger i github
* Satt opp hosting av postgress SQL database gjennom platformen Neon som er lett tilgjengelig via vercel.
* Laget røff skisse for SQL i mySQLworkbench for å ha en skisse å vise til i planlegginsdokumentet.
* Lage JSON struktur som skal lastes opp som fil
* Skisse for oppkobling mellom bruker av nettsiden og server hvor websiden hostes og hvor databasen er hostet
* Skrevet ned flere ideer om hvordan produktet evt kan forbedres hvis tiden tillater det’

Dag 2 – Første arbeidsdag

* Lagde enkel skisse for utgangspunkt for design til websiden
* Lagde tabeller i PostgreSQL databasen basert på skissen
  + Utvidet «team» tabellene med en foreign key fra «Quizez»
* Påbegynnt backend kode
  + La inn admin bruker
  + Støtte på feil underveis som ikke var tydelige
    - Støtte på feil hvor ‘fs’ ikke var mulig å finne for en tilfeldig node module jeg ikke hadde endret på
    - Tror feilet oppstå i en rar rekke, hvor flere småfeil med next.js jeg hadde gjort sammen fikk feilen til å skje.
* Lagde header komponent
* Backend kode som brukes til loginn for admin
  + Funksjonen for loggin setter verdier inn i session
* Har brukt irriterende tid på en modal
* Fått klar modal som kan brukes til loginn for admin, og når admin laster inn fil.
* Litt utseende til loggin for admin inne i modalen laget
  + Ingen funksjonalitet, men burde være lett når den knyttes til funksjonen som er laget.

Notat før dagen,

Kan lages komponenter av input felt og knapper for å få koden ryddigere

Dag 3- andre arbeidsdag

* Under oppretting av admin bruker har verdiene som er knyttet til admin brukers loginn blitt pushet ut til git, det er for å vise prosessen som jeg gikk gjennom for å opprette brukeren. I en ordiner prosess ville denne feilen ha blitt fikset med å fjerne commiten fra git.
* Støtte på et problem med at en variabel ble sendt inn som undefined, selv om den hadde en verdi. Tror det var på grunn av rekkefølgen koden kjørte
  + Modalen som ble plassert inn i koden gjennom en en setState(), tror jeg bare fikk en versjon av Loginn() funksjonen mens variablene ikke hadde fått verdi fra input feltene. Under loginn for admin.
* Fikset masse småfeil ved loginn funksjoner, som at session ble ikke lagret derfor ble ikke satt og middleware routet meg tilbake til startsiden så det så først ut som at router.push() hadde ingen effekt.
* Fil innlesning av JSON fil funker til websiden, begynner å lage kode som legger Quizer inn i databasen.
* For mange ledd til koden som blir laget så det er vanskelig å få et godt overblikk over hvor dataene går galt
  + Fikset alle småfeil, nå går det greit for admin å lese JSON fil inn i databasen fra web.
* Påbegynt frontend listing av quizer

Dag 4 – tredje arbeidsdag

* Quiznavn og beskrivelse blir listet opp med dataene som hentes fra databasen
  + Har arbeidet med dette i stor grad og med masse små led som henger sammen for at det skal funke
  + Har støtt på en del små problemer underveis, kunne ha blitt unngått hadde jeg tatt en mer direkte vei for å teste biten som henter quizen fra databasen og som blander om rekkefølgen på spørsmålene og kategoriene hentet fra databasen. Istedenfor å prøve å lage frem løpet som skal være funksjonelt for websiden.
* Dataen fra databasen har jeg fått hentet ut i formatet jeg ville ha som da lar meg begynne på å vise frem quizen til brukeren, og da gjenstår og legge svarene til Lagnavnet inn i databasen og at admin kan se resultater til hver quiz. Osv etter det kan Tvist og eventuelle forbedringer og finpuss bli sett på
* Logikk gir mening så langt jeg har kodet den nå for fremvisning, må teste i morgen.

Dag 5 – fjerde arbeidsdag

* Fremvisning av quiz til bruker funker
  + Blir vist frem til bruker tilfeldig hver gang de går inn på quizen
  + Dataen blir hentet ut i tilfeldig rekkefølge frå databasen via ORDER BY RANDOM()
* Brukerens svar blir gjort om til en liste som serversiden skal gjøre om til rader i answers tabellen i databasen når brukeren svarer på siste spørsmål
* Laget SQL for å sette inn team og hvilken quiz de tar, i tabellen teams.
  + Returnerer teamID som blir brukt når svarene til brukeren blir loopet over og satt inn i databasen.