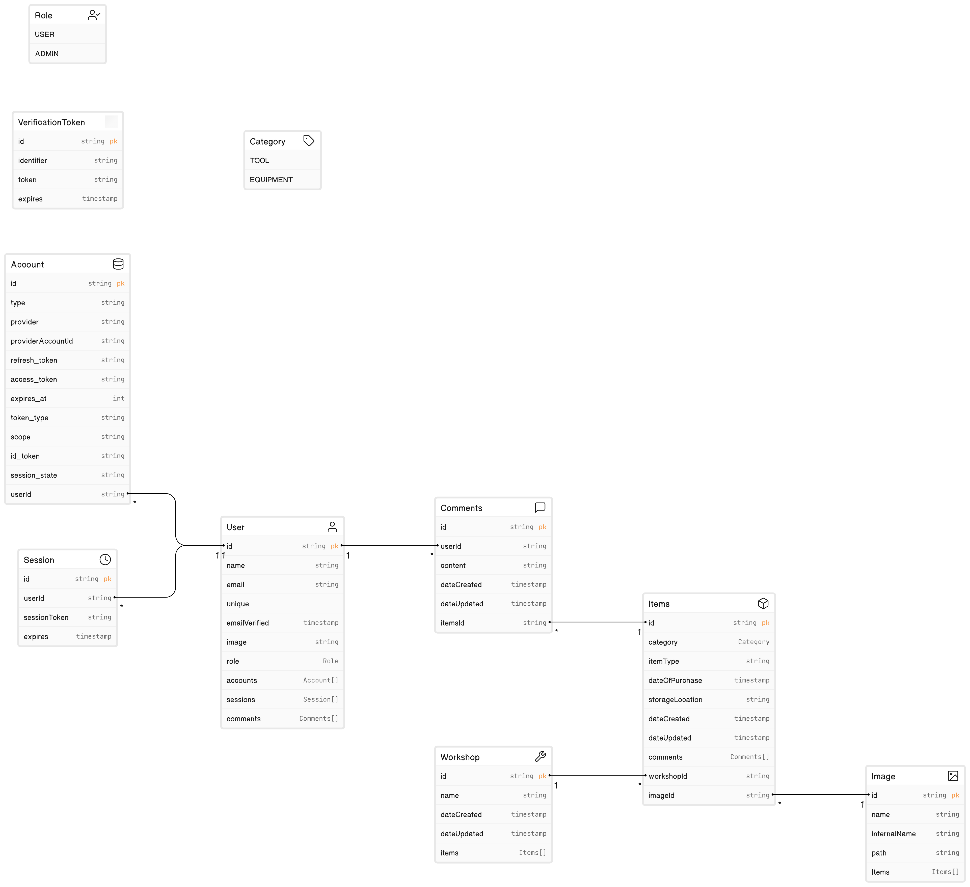
# Logg

## 17:15 30.05.2023

I dag har jeg fikk jeg godkjent på prosjektplanen for arbeidet jeg skal utføre. Etter at jeg fikk godkjent satte jeg i gang med arbeidet mitt. Det første jeg gjorde var å sette opp et nytt NextJs prosjektet og deretter laste opp prosjektet til GitHub som «grunnleggende» kode. Det neste jeg gjør er å opprette en BackLog til prosjektet hvor jeg skriver diverse problemstillinger og egenskaper som skal legges til. Fram til nå har jeg utført alle 4 oppgaver jeg skrev opp som er #1: Opprette Navbar, #2: AuthJs, #3: Sette opp prisma og planlegge database tabeller og #4: Oppsett av Database. Jeg har valgt å sitte lengere å jobbe med dokumentasjonen.

Skisse av database (Skapt av eraser.io basert på mine prisma modeller)



Skisse av database (Laget Manuelt)

Et bilde som inneholder tekst, skjermbilde, diagram, nummer

Automatisk generert beskrivelse

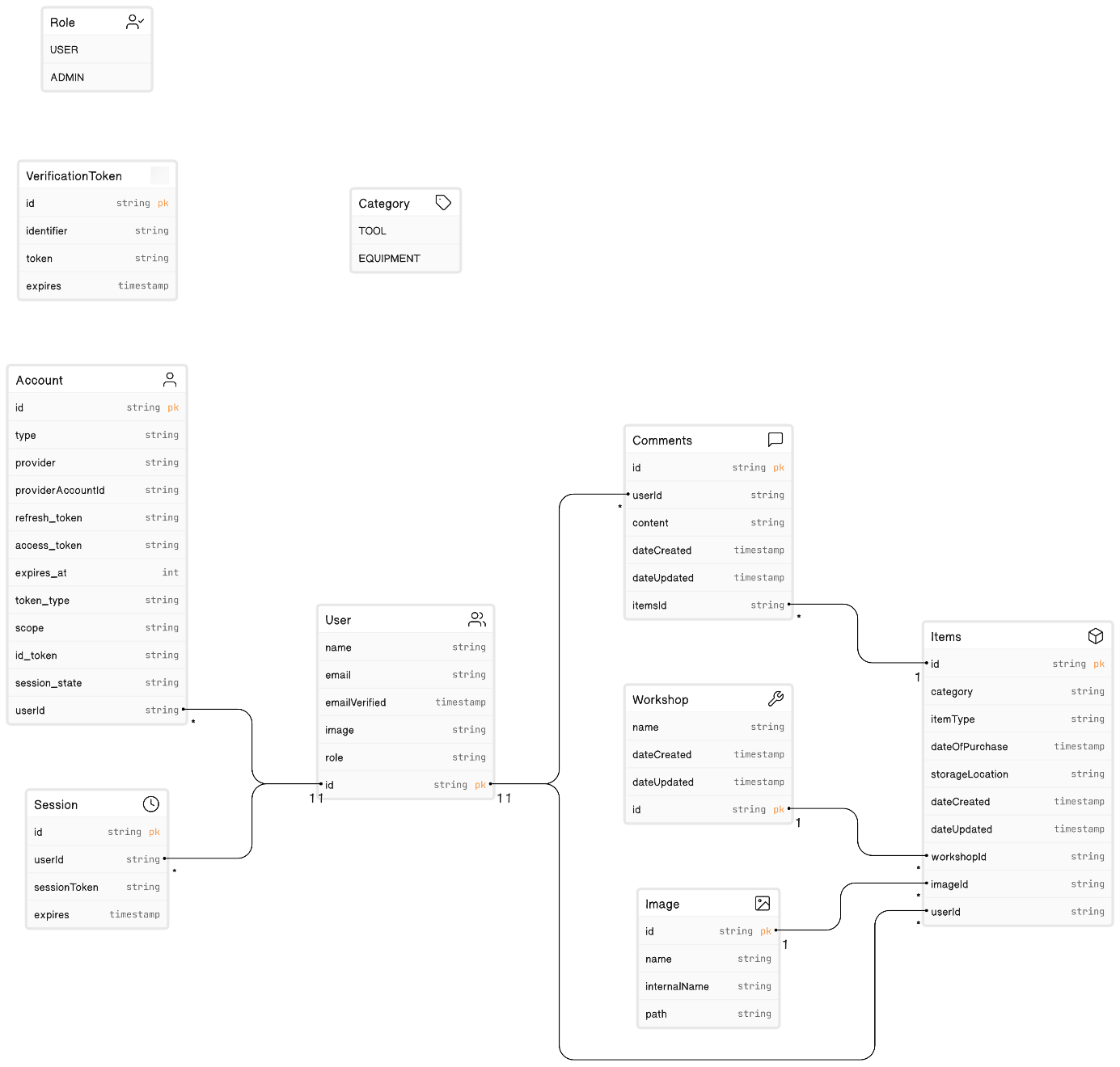
19:15 Ferdig for dagen.

Hjelpemidler brukt i dag:

* Har fått litt råd fra kollega at jeg nesten gjorde en feil angående relasjon mellom tabellene «Items» og «Image».
* Har brukt Figma til å manuelt lage en visuell struktur av tabeller
* Har fikk tips fra kollega i å bruke eraser.io til å skape visuell struktur av tabeller (AI)

## 09:40 31.05.2023

Jeg tenker å legge til userId i tabellen «Items». Jeg tenker at det kan være gunstig å ha muligheten til å identifisere hvem som oppretter et nytt verktøy og utstyr som skal vises på nettsiden. I oppdraget som er delt ut står det at Admin skal kunne legge til eller slette verktøy, så jeg kunne fortsatt ha løst dette uten bruken av userId.

Skisse av oppdatert database (Skapt av eraser.io basert på mine prisma modeller)

Skisse av databse (Laget Manuelt)

Et bilde som inneholder tekst, skjermbilde, diagram, nummer

Automatisk generert beskrivelse

10:49 laster ned Zod for å analysere req.query.itemId i Delete Route for sletting av Item.

Under på bildet kan vi se at itemId kan enten være string, string[] eller undefined.

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

Planen med å bruke Zod er for å parse req.query.itemId til at itemId bare er string.

Løsning:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

15:27

I dag har jeg brukt dagen til å sette opp routes for Items, workshop og comment.

Et bilde som inneholder skjermbilde, tekst, Font

Automatisk generert beskrivelse

Jeg har også lagt til test-data som jeg kan bruke under testingen.

Nå henter jeg dataen fra databasen med bruk av SWR Hook. Da jeg satt opp en .get route for Item, så tenkte jeg å hente ut alle verktøyene og utstyrene direkte fra tabellen, men etter å ha skrevet ferdig den routen, tenker jeg heller at det er mulig å bare hente ut workshop og deretter inkludere Items-tabellen ilamme workshop.

Logikken er, hente workshop fra «workshop» tabellen med bruk av SWR Hook -> inkludere «Item» tabellen i samme sleng. Prisma tillater å inkludere tabeller som har relasjoner med hverandre, så da kan jeg lett hente alt fra samme spørring istedenfor å ha flere.

Bilde av løsning:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Dette får meg til å tenke at jeg hovedsakelig ikke trenger å ha tabell for «workshop», men jeg velger å ha den for å gi navn på verkstedet og kan gi frihet for å beskrive verkstedet.

19:41 Ferdig for dagen

Hjelpemidler brukt i dag:

* Har sett på tidligere kode om hvordan jeg satt opp SWR Hook og sett på dokumentasjonen demmes

## 14:47 1.06.2023

I dag har jeg strevd med Pagination i nesten hele dag fordi useSWR vil ikke oppdatere verktøyene øyeblikkelig når jeg blar til side 2 eller 3. Derimot så oppdaterer dataen seg etter jeg har klikket ut av nettsiden og tilbake (focus page reload). Har fått litt hjelp av kollega, men vi begge klarte ikke å helt finne ut hvorfor. Kollega måtte til et viktig oppmøte, så jeg valgte å prøve videre selv fram til nå. Har nå bare valgt å heller vente med dette til senere eller når han er tilbake.

På morgenen brukte jeg også 1 time på å verifisere databasen. I mine øyne så ser databasen riktig ut etter det jeg har sett for meg, men har ikke enda testet alle tabeller. Tabellen jeg ikke har testet er «Image» som jeg skal bruke for å lagre bildene’s sti og navn for hvert verktøy og utstyr. Jeg tenker også å kanskje legge til beskrivelse og navn til tabellen «Items» for mer og bedre beskrivelse på verktøy og utstyr.

16:10

Nå har jeg fått hjelpen jeg trengte og laget ferdig pagination for verktøy og utstyr.

Bilde av pagination:

Et bilde som inneholder tekst, skjermbilde, Font, design

Automatisk generert beskrivelse

Har ikke fått laget utseende enda, men det vi ser her er verktøyTypen for hvert verktøy og utstyr i databasen. Som vi ser så kan man nå bla fra side 1 til 3.

Hjelpen jeg trengte for dette var med hjelp fra min kollega og ChatGPT (for å regne sidetall og antall sider).

Tidligere nevnte jeg at dataen ikke ville oppdaterte i sanntid, men min kollega fant ut at løsningen for dette er å bruke useEffect i funksjonen hvor jeg sender ut dataen. Dataen jeg sender ut fra fetchHooken «GetItems» er verktøy og utstyr, error, is loading og mutate funksjonen.

Som vist på bildet under



Items er et array med alle utstyrene og verktøyene slått sammen, mens error og mutate er standard fra useSWR. Mutate lar meg som tidligere sagt oppdatere dataen når mutate funksjonen er kjørt.

På samme side hvor jeg viser frem dataen, prøvde jeg å kjøre mutate() enten i selve skriptet (som en test først), ved onClick når jeg skulle bla frem og tilbake på sidene osv.

Løsningen er nemlig at jeg må kjøre useEffect inni «GetItems» funksjonen hvor jeg henter dataen.

Tidligere så hadde jeg allerede lagt til funksjonene nextPage og prevPage som setter antallet verktøy som skal hentes.

Bilde av «GetItems» funksjonen

Et bilde som inneholder tekst, skjermbilde, programvare

Automatisk generert beskrivelse

På slutten av useEffect så ser man at jeg har lagt til pageOffset, pageLength og mutate. Logikken er at med en gang pageOffset og pageLength staten blir oppdatert, så vil useEffect kjøres. Da kjører jeg nemlig mutate funksjonen inni der. Det vi da få dataen til å oppdatere seg i sanntid på nettsiden.

Da det var gjort, så gjenstod bare knappene som lar oss bla fra side til side. Jeg ville nemlig at en av dem skulle fjernes dersom det ikke finnes flere sider fremover eller bakover

Det første vi måtte gjøre var å hente totalt antall «items» fra databasen. Prisma har en funksjon som lar meg hente ut totalt antall som vist på bildet under:



Det neste vi gjorde da var å returnere ut item og totalAmount som et objekt:



Deretter måtte vi lage en ny type som tilsvarer dataen vi får fra databasen

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Og til slutt returnere pageLength, pageOffset, items og count tilbake til client.

Et bilde som inneholder tekst, skjermbilde, Font, design

Automatisk generert beskrivelse

For å ordne slik at knappene forsvinner, brukte jeg bare en && operator:

Et bilde som inneholder tekst, skjermbilde, programvare

Automatisk generert beskrivelse

Bilde fra da jeg spurte ChatGPT:

Kalkulere hvilken side man er på

Et bilde som inneholder tekst, skjermbilde, programvare, Multimedieprogramvare

Automatisk generert beskrivelse

Kalkulere totalt antall sider

Et bilde som inneholder tekst, skjermbilde, programvare, Font

Automatisk generert beskrivelse

20:39 Ferdig for dagen

Hjelpemidler brukt i dag:

* Kollega
* ChatGPT
* Sett på eksempler fra tidligere kode
* Utseendet for createItem er hentet fra tailwindUI. <https://tailwindui.com/components/application-ui/forms/form-layouts>

9:46 02.06.2023

La til navn på Item modellen i prisma.schema. Velger å legge til navn for både verktøy og utstyr. Grunnen for at jeg ikke la det til tidligere med en gang var fordi jeg tenkte å bruke «itemType» som beskrivelse når brukere ser verktøyet på nettsiden.

10:49 Viste fram innleggingssiden til kollega da han gikk forbi. Jeg fikk litt styling-tips og la til en DatePicker.

DatePicker: <https://www.npmjs.com/package/react-datepicker>

Jeg vil at administrator skal få velge dato da verktøyet ble kjøpt, så denne DatePicker’en er perfekt for dette.

12:06 Ferdig med utseendet for innleggingssiden

12:28 Min kollega nevnte en viktig del som er at jeg ikke hadde lagt til sikkerhet på backend enda. Jeg hadde nemlig ikke lagt til en funksjon som sjekker om brukeren er logget inn eller er administrator når man skal legge til nye verktøy og kommentarer. Jeg hadde også lagt til «userId» på verktøyene og sender dem via req.body, som man egentlig ikke skal gjøre, men er mulig. Grunnen for at man egentlig ikke skal gjøre det er for å hindre andre brukere/hackere i å få tak i ID’en til enhver bruker når post, put og delete requests blir utført.

For å fikse opp i dette så måtte jeg hente brukerens ID fra session.

Sammen så lagde vi en middleware med bruk av funksjonen router.use() som skal verifisere om brukeren er logget inn, om brukeren er Admin og om brukeren er skaperen av verktøyet som skal slettes eller oppdateres. Når det gjelder å opprette nye verktøy, så trenger vi bare å sende inn «userId» som skal legges på verktøyet i databasen.

Det første vi gjorde for å løse dette var å lage en ny type hvor vi legger user.session på req.

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Deretter la vi til router.use() over delete og put rutene.

Et bilde som inneholder tekst, skjermbilde, programvare

Automatisk generert beskrivelse

Jeg sjekker nå altså om brukeren er innlogget, om brukeren er Admin og om brukeren eier verktøyet.

For å hente «userId» fra session så bruker jeg funksjonen getServerSession som er fra AuthJs sin dokumentasjon.

Vi lagde først en test i [itemId].ts hvor vi brukte getServerSession og la på user.session i req, men flyttet dette og lagde det som et middleware for å gjenbruke dette for alle API rutene.

Bilde av getUser middleware:

Et bilde som inneholder tekst, skjermbilde, programvare

Automatisk generert beskrivelse

Nå trenger jeg ikke å sende «userId» via req.body for å legge til ID’en til enten verktøy eller kommentarer.

Før endringen:

Et bilde som inneholder tekst, skjermbilde, display, Font

Automatisk generert beskrivelse

Etter endringen:

Et bilde som inneholder tekst, skjermbilde

Automatisk generert beskrivelse

16:20 Jeg og min kollega så på hvordan man kan fremvise bildet før det lagres på nettsiden. Vi fant en løsning som viser hvordan det settes opp i react.

Link til løsning: <https://stackoverflow.com/questions/38049966/get-image-preview-before-uploading-in-react>

Link til Toastify: <https://www.npmjs.com/package/react-toastify>

Formidable Løsning: <https://stackoverflow.com/questions/30128701/parse-form-value-with-formidable-to-filename>

Har brukt formadible veldig lite for håndtering av bilder, så velger å hente et tidligere eksempel fra et tidligere prosjekt.

Prosjektet jeg henter koden fra er et privat prosjekt som jeg har på min GitHub. Koden er som følger:

/utils/formadible-async.ts

Et bilde som inneholder tekst, skjermbilde, Font, display

Automatisk generert beskrivelse

/api/image/index.ts

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

21:06

Fram til nå har jeg strevd med opplastningen av bilder fordi jeg fikk en feilmelding fra axios som hadde meldingen: AxiosError: Request failed with status code 500. Jeg forsto ikke hva det gjaldt, men sjekket terminalen på serveren om hva som kunne ha skjedd. Da jeg sjekket i terminalen, sto det: TypeError: resolver is not a function. Jeg så in på det og søkte opp en løsning på nett, men fant ikke en løsning. Jeg satt deretter opp routen i API’et på nytt for å sakte men sikkert teste om dataen når serveren. Dette fungerte selvsagt etter 2 timer med feilsøking..

Annet: Jeg har endret emum for Category i prisma.schema fordi tidligere så hadde jeg definert kategoriene for «TOOL» og «EQUIPMENT». Endret til norsk «VERKTOY» og «UTSTYR» fordi jeg ikke tenkte tidligere. Allikevel så har jeg lagt inn data med kategorien «TOOL» og «EQUIPMENT», så jeg kaller dette for en «blind bevissthet».

Før:

Et bilde som inneholder Font, tekst, skjermbilde, Grafikk

Automatisk generert beskrivelse

Nå:

Et bilde som inneholder Font, tekst, Grafikk, skjermbilde

Automatisk generert beskrivelse

Tidligere hvis tabellen «Items» var tom for verktøy og utstyr, så ville en if statement returnere dersom det var tomt og man hadde ikke muligheten for å legge til verktøy fordi pluss knappen ikke vistes opp. Nå har jeg lagt til knappen sammen med teksten «laster data…» dersom databasen er tom.

Før:



Nå:

Et bilde som inneholder tekst, skjermbilde, Font, display

Automatisk generert beskrivelse

21:54 Ferdig for dagen

03.06.2023 18:10 Nå har jeg forbedret litt på utseendet i filen item.tsx

Slik ser det ut nå:

Et bilde som inneholder line, skjermbilde, tekst

Automatisk generert beskrivelse

Istedenfor å hente «imageId» sammen med verktøyet, så har jeg endret dette til «internalName» fra image. Jeg trenger nemlig navnet istedenfor ID’en.

Før:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Etter:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Jeg la også på «internalName» på typen IItem:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Deretter la jeg til stien på elementet <Image />:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Det som gjenstår, er å lage en GET Route il filen /api/image/[id].ts somhenter bilde-stien hvor bildet ligger (/public/img).

Jeg har allerede skrevet dette tidligere, så jeg velger å gjenbruke tidligere kode for dette.

Bilde av hentet kode:

Et bilde som inneholder tekst, skjermbilde, programvare, display

Automatisk generert beskrivelse

Det som skjer først, er at navnet til bildet blir sendt til API’et og mottat av denne filen fordi filen heter [id].ts

Når filer har [ ] (Square Brackets) rundt seg, så leser/mottar tekst som blir sendt til stien. For eksempel /api/image/**23817313128937.png ->** leste **23817313128937.png.**

Jeg henter teksten med bruk av req.query og legger teksten i variabelen «id»

Videre så sjekker jeg om variabelen er av typen string:

Et bilde som inneholder Font, tekst, skjermbilde, line

Automatisk generert beskrivelse

Deretter henter bildet hvor navnet tilsvarer samme navn som er i variabelen:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Jeg bruker fs.readFileSync som er en Node Module som lar meg lese bildet. Bildet inneholder masse tall som ser slik ut:



Nå vises bildet på nettsiden:

Et bilde som inneholder skjermbilde

Automatisk generert beskrivelse

21:35 Ferdig for dagen

12:30 Innser at jeg heller burde ha generert en egen id for gjenstander slik at jeg kan bruke den istedenfor «ItemId». Det er med tanke på sikkerhet og for å hindre «ItemId» til å vises i URL’en.

Klokken 18:35 loggfører jeg at jeg har brukt ChatGPT til å generere utseendet for kommentarsiden. Grunnen for dette er for å spare tid.

Bilde av kommentarsiden:

Et bilde som inneholder verktøy, skrutrekker

Automatisk generert beskrivelse

Endret en tidligere custom hooks som het GetWorkshopWithItem.ts til GetItemWithComments.ts

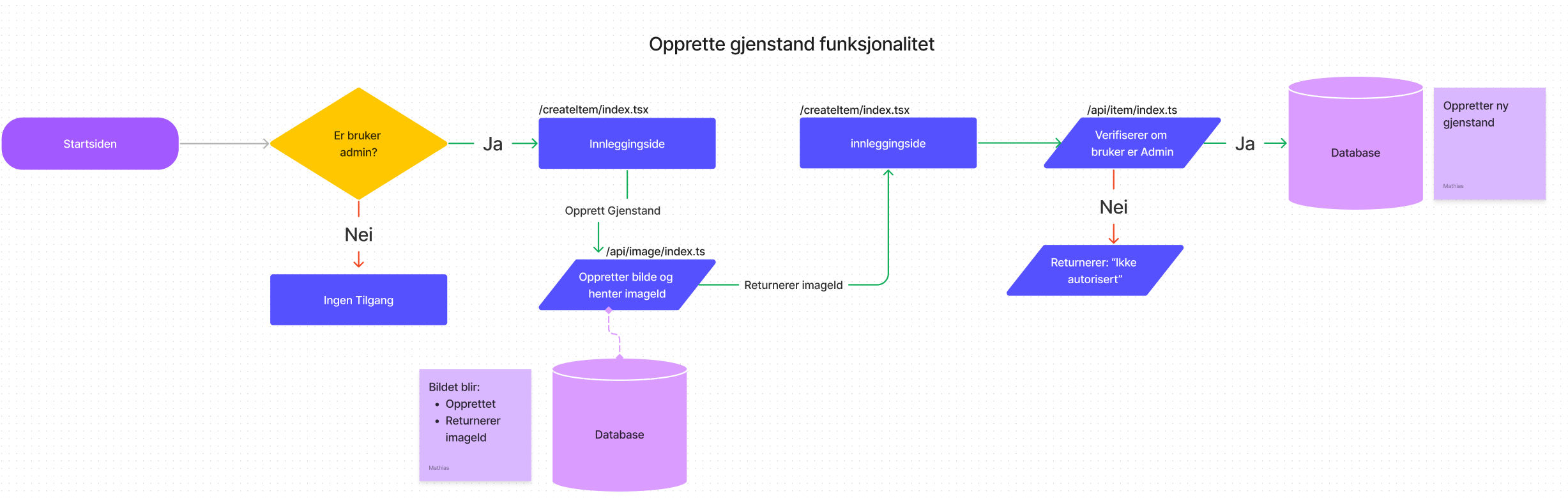
Et bilde som inneholder tekst, skjermbilde, display, programvare

Automatisk generert beskrivelse

Endret og la til nye types:



Laget skisse for funksjonalitet:



Et bilde som inneholder skjermbilde, tekst, diagram, Font

Automatisk generert beskrivelse

Link til diagram: <https://www.figma.com/file/ldxisHJwlIw2JJ2yTHoXac/Untitled?type=whiteboard&node-id=0-1&t=0OinnVa0hOypKB9B-0>

Flyttet filer og mapper og gitt beskrivende navn.

Opprettet en egen side for en gjenstand:

Et bilde som inneholder tekst, skjermbilde, Font, design

Automatisk generert beskrivelse

Koden fra siden:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Opprettet en ny fil i API ’et som henter gjenstander med kommentarer og bruker:

Et bilde som inneholder tekst, skjermbilde

Automatisk generert beskrivelse

Har laget en egen komponent som fremviser kommentarer:

Et bilde som inneholder tekst, skjermbilde, programvare

Automatisk generert beskrivelse

Komponenten over blir kjørt fra en ny fil som heter SingleItemPage.tsx som også er en komponent.

Første del av singleItemPage.tsx:

Et bilde som inneholder tekst, skjermbilde, display, programvare

Automatisk generert beskrivelse

Har lagt til en «form» hvor bruker kan opprette kommentarer på et verktøy og utstyr.

Andre del av singleItemPage.tsx

Et bilde som inneholder tekst, skjermbilde, programvare, display

Automatisk generert beskrivelse

Utsendet er laget av ChatGPT og jeg har modifisert litt av den.

Bilde av hentet kode:

Et bilde som inneholder tekst, elektronikk, skjermbilde, programvare

Automatisk generert beskrivelse

Generert av ChatGPT fordi jeg ville spare tid og jobbe videre med funksjonalitet.

Det siste jeg jobber med er funksjonaliteten for å slette kommentarer via client. Har hatt problemer med å motta «commentId» i API ‘et fordi jeg får et problem med zod.

Følgende feilmelding:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

22:05 Ferdig for dagen

Lenker brukt i dag:

ChatGPT: <https://chat.openai.com>

TailwindCSS: <https://tailwindcss.com/>

05.06.2023 15:41

Loggfører nå at jeg fikk tips av å sjekke et sikkerhetsproblem som var å åpne nettsiden i edge og gå til localhost:3000/api/item/ 647dddbd61e806f185460cfc og se om verktøyet vises med kommentarer eller ikke. Det gjorde nemlig det, så fant jeg ut at jeg hadde glemt å legge til en sjekk på denne API-stien som sjekker om brukeren er innlogget eller ikke.

Fikset problemet med denne if-statement i /api/item/[itemId].ts:



Jeg fikk også tips om å fikse mappestrukturen min fordi det så rotete ut. Jeg var også enig i måten jeg satt opp mappestrukturen jeg satt opp i går kveld ikke så så bra ut, så jeg endret dem og fikk en «a-ha» opplevelse da jeg så at jeg kunne bare ha laget GET Routen i /api/item/[itemId].ts istedenfor å ha /api/item/comment/itemWithComment/[itemId].ts

Flyttet også mappen /comments/ ut fra /item/, så nå ser mappestrukturen mye bedre ut enn tidligere.

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Jeg har fått testet nettsiden som vanlig bruker, som admin og med andre brukere takket være hjelp fra kollega. Jeg fikk senere lagt til en sjekk som ikke tillater brukere som ikke er innlogget til å se kommentarer og brukere som har skrevet en kommentar har mulighet til å se sletteknappen på sin kommentar.

Et bilde som inneholder tekst, skjermbilde, display, Font

Automatisk generert beskrivelse

Jeg har klart å få slettefunksjonen for kommentarer til å fungere. Feilen jeg gjorde var at jeg hadde skrevet req.query.itemId istedenfor req.query.commentId

Feil:

Et bilde som inneholder tekst, Font, skjermbilde

Automatisk generert beskrivelse

Løsning:

Et bilde som inneholder tekst, skjermbilde, Font, line

Automatisk generert beskrivelse

Har laget skisse for funksjonaliteten som henter gjenstandene fra databasen:

Et bilde som inneholder tekst, skjermbilde, diagram, Font

Automatisk generert beskrivelse

Skisse for sletting av kommentarer:

Et bilde som inneholder skjermbilde, tekst, diagram, Font

Automatisk generert beskrivelse

Skisse for oppretting av kommentarer:

Et bilde som inneholder tekst, skjermbilde, diagram, Post-it-lapp

Automatisk generert beskrivelse

Skisse for henting av gjenstand med kommentarer og skaperen av kommentaren:

Et bilde som inneholder tekst, skjermbilde, Font, diagram

Automatisk generert beskrivelse

13:13 06.06.2023

Skulle vise en bug til kollega, men fant en annen hvor det skjer 8 – 10 spørringer til databasen for å hente gjenstander. Fikset buggen i filen components/items/item.tsx. Problemet var at jeg valgte å importere custom hooken GetItems.ts, og kjøre «mutate» funksjonen istedenfor å sende funksjonen «mutate» via komponenten.

I filen /pages/index.tsx

Før:

Et bilde som inneholder tekst, skjermbilde, Font, line

Automatisk generert beskrivelse

Etter:

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

I filen /component/items/item.tsx

La til type

Et bilde som inneholder tekst, skjermbilde, Font, display

Automatisk generert beskrivelse

17:04 har dokumentert mye og skrevet ned beskrivelse om valgte moduler.

Har nå også laget ferdig en skisse som viser nettsiden når man ikke er innlogget, innlogget og innlogget med Admin rettigheter:

Et bilde som inneholder tekst, diagram, skjermbilde, plan

Automatisk generert beskrivelse

20:52

Fram til nå har jeg dokumentert mye, lest igjennom dokumentasjonen sammen med kollega og rettet skrivefeil.

Systemskissen er også skrevet ferdig. Nå gjenstår bare egenvurdering dokumentere dagens arbeid i heftet dokumentasjon Arbeidsprosessen og lage en kjapp «setup» veiledning på GitHub i filen README.

22:29

Har skrevet README nesten ferdig, har startet på egenvurderingen og nesten skrevet ferdig Dokumentasjon Arbeidsprosessen. Har også lastet opp word dokumentene på GitHub i prosjektet.