Dokumentasjon

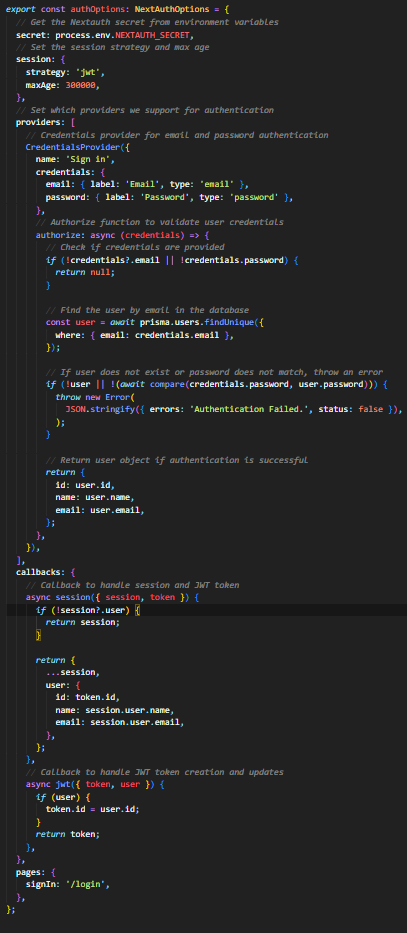
Mathias Voie Johansen – 06.02.25

Zod

A screen shot of a computer program

AI-generated content may be incorrect.Ble brukt for validering av forms på klient side og server side for å forsikre at dataen som blir sendt til serveren er trygg. Modulen fjerner også stress på serversiden siden den stopper api kallet om den forstår at dataen som blir postet ikke er OK.

NextAuth

Nextauth er en autentiseringsmodul som har mange integrerte tjenester du kan lett sette opp autentisering mot. Som google, github og discord osv... Dem gjør det også lett å integrere løsningen til alle prosjekter/løsninger.

Har satt max age til 3,5 dager som vil si at når noen logger inn holder de seg innlogget i 3 og en halv dag for å forsikre seg at ikke noen kommer flere uker/år etterpå å fortsatt har tilgang til siden.

I authorization sjekker vi om alt er definert. Om ikke returnerer vi null. Så henter vi brukeren ved hjelp av epost addressen som var oppgitt. Om brukeren eksisterer sammenligner vi det hashet passordet med bcrypt. Om passordet stemmer og brukeren eksisterer oppretter vi session til brukeren med dataen som vi velger å returnere (id, name & email)

Routing

A screen shot of a computer

AI-generated content may be incorrect.NextJS hat et helt ‘perfekt’ men avansert innebygd system for routing med alt fra ‘Parallell routes’, ‘Intercepting routes’ og ‘Dynamic routes’ for å tilfredsstille brukere og utviklere. I prosjektet ble det brukt intercepting routes for modal håndtering. Dette vil si at på selve siden trenger jeg ikke å håndtere logikken om modaler er åpen eller ikke siden de er sekludert i en helt annen sektor i koden.

Intercepting routes blir også automatisk importert til layouts (I samme directory) som da er hvorfor alt klarer å høre sammen. Det gjør det også veldig enkelt for rollestyrt sider.

Tailwind

Tailwind lar meg bruke premade klasser for jeg kan lett sette på HTML objekter for å få dem til å se ut akkurat som jeg vil uten å røre en .css fil som gjør jobben for meg som utvikler mye lettere.

Prisma

A screen shot of a computer code

AI-generated content may be incorrect.Prisma lar deg lett sette opp intellisense med databasen din ved å bruke en prisma.schema fil. Lar deg også kommunisere lett med relaterte databaser med a include query.

Her ser man schemaet for Verktøyene som da inneholder imageId og folderId for å lage en relasjon til hvordan mappe dem tilhører. Her kan også annen data som vi vil skal dokumenteres på hvert verktøy puttet inn.

A screen shot of a computer program

AI-generated content may be incorrect.

På bunnen sier vi også at det er en unique constraint på id og folder id. Som vil si at det ikke kan være duplicate verktøy satt til samme mappe.

Tidsplan

Her er en oversikt over planen jeg fikk godkjent av prøvenemda også har jeg markert med grønn og rødt av endringene som ble gjort underveis.

**Mandag**

* Lese og forstå oppgaven som er levert.
* Lage løsningsforslag
* Jobbe med den teoretiske delen av oppgaven
* Klargjøre GitHub repository.
* Oppsett backend (PostgreSQL)
* Sette av tid til egenvurdering og dokumentasjon

**Tirsdag**

* API ( POST, GET)
* Rolle-styrt adgang
* Bildeopplastning til server
* Autentisering
* Sikkerhet for brukere
* Sette av tid til egenvurdering og dokumentasjon

**Onsdag**

* API ( PUT, PATCH, DELETE )
* Bildeopplastning til server
* Utforming & design av nettside
* Utvikle forms til forskjellige handlinger
* Sikkerhet for brukere
* Sette av tid til egenvurdering og dokumentasjon

**Torsdag**

* Putte formene som ble utvikler på onsdag i modals for lett tilgjengelighet
* Ferdigstille bildeopplastningen.
* Sikkerhet for brukere
* Sette av tid til egenvurdering og dokumentasjon
* Utvikle forms til forskjellige handlinge
* Utforming & design av nettside

**Fredag**

* Sikkerhet for brukere
* Ferdigstille bildeopplastningen.
* Ferdigstille utviklingen av API
* Ferdigstille kommunikasjon mellom frontend og backend
* Sette av tid til egenvurdering og dokumentasjon

**Mandag**

* Vurdere HMS i henholdt til oppgaven
* Få prosjektet online med SSL sertifikat.
* Ferdigstille dokumentasjon og egenvurdering
* Sette av tid til egenvurdering og dokumentasjon

**Tirsdag**

* Presentere fagprøven for prøvenemda

Fjernet punktet med å få prosjektet online med SSL sertifikat siden jeg ikke følte det var relevant for vurdering.

HMS

Under oppgaven var HMS en stor prioritet fra god luftkvalitet til ergonomisk stol for en veldig god sittestilling. Tid for lufting av hodet, frisk luft og trening ble også utført under oppgaven.

Daglig arbeid

**Mandag 26.05.25:**

Fikk utlevert oppgaven kl. 13:00 og startet med å vurdere ulike prosjektideer. Etter en kort ide laging bestemte jeg meg for å utvikle en digital verktøykasse. Sattt også opp test databasen som skulle brukes under utvikling. Deretter lagde jeg en plan for de neste dagene, inkludert oppgaver som kravspesifikasjon, valg av teknologier, oppsett av utviklingsmiljø, samt design og implementasjon av brukergrensesnitt og funksjonalitet. Begynte også på kvelden å lage en skjelett som jeg skulle følge framover.

**Tirsdag 27.05.25:**

Fortsatt der jeg slapp dagen før med videreutvikling av skjelettet på siden. Begynte på skjelettene til modalene uten funksjonalitet. Fikk også ned autentiseringen med nextauth med innloggingsside og registreringsside som lagrer og benytter brukerinformasjonen på en effektiv og sikker måte. Begynte å lage bildeopplastning siden jeg følte det var noe jeg måtte prioritere, brukeren opplaster filen til backendet også gjør serveren det om til binær før det blir klargjort til lagring på serveren.

**Onsdag 28.05.25**

Starter utviklingen av API slik at modal formene har et backend dem kan kommunisere med. Lager også et design som er naturlig, tilgjengelig og appelerende for alle. Dette er da en form for design jeg fulgte hele veien igjennom prosjektet.

**Torsdag 29.05.25**

Videre utviklet og ferdigstillte formsene i modalene for å kommunisere med backend ved bruk av axios. Ferdigstillte også bildeopplastningen slik at den er trygg og sikker for trafikk fra klienter.

**Fredag 30.05.25**

Ferdigstillte utviklingen av API og kommunikasjonen fra klienten til serveren. I dag tok jeg meg et steg tilbake for å få en liten puste pause og reflektere skikkelig over valg som er blitt gjort de siste dagene. Da begynte jeg å ekspirimentere med js classes for å se om det er noe som kan effektivisere framgangen i prosjektet som var en fantastisk ide. Gikk tilbake og endret ting til å bruke klassene jeg lagde. Ble mye bedre orden i koden.

**Mandag 02.06.25**

Videre utviklet & kommenterte / ryddet opp i koden som jeg har produsert de tidligere dagen for at det skal være mere lesbart for folk som ikke har holdt på med prosjektet. Ferdigstilt alt av kode og dokumentasjon slik at det bare er fremføringen som gjenstår.

**Rolle styrt autentisering**

**A white rectangular object with black text

AI-generated content may be incorrect.**

Når brukeren er innlogget og er administrator kan man se forskjellige actions som man kan utføre på siden. Dette er gjort på serversiden under compiling av siden. Også replikert på api endepunktene for maksimal sikkerhet.

**Modals**

A screenshot of a chat

AI-generated content may be incorrect.Med bruk av en component som HeroUI har gjort lett tilgjengelig tilfredsstiller utseende og animasjonene brukeren. Det er også lett å navigere igjennom dem for å få ønsket resultat.

**Activation**

Måten jeg har valgt for å finne ut om modalen skal være aktiv eller ikke er å bruke searchparams.

Grunnen for dette valget er at hvis en bruker skal vise noen som sitter på en annen pc hva de ser, og de sender linken til siden. Vil modalen og alle instillingene deres komme opp på skjermen til brukeren de skal vise det til.



Altvernativet kunne vært å bruke useState men jeg synes det blir for gammeldags med valg som er lagret i memory istedenfor i url når det kommer til nettsider.

Her er da måten jeg henter ut searchParams på. Veldig enkel og fin måte også har jeg en isOpen variabel som da sjekker om modalen skal være aktiv eller ikke.

A screen shot of a computer code

AI-generated content may be incorrect.

isOpen verdien kan også passeres til Form komponenten hvis du skal hente data som er relevant til modalen slik at de ikke blir fetchet med en gang du kommer inn på siden men når du spør etter den dataen.

Database

A screen shot of a computer program

AI-generated content may be incorrect.Her ser du en oversikt over schemaet jeg har laget for databasen. Ved videreutvikling kan man legge ved så mange punkter / informasjon man har lyst på mappene og verktøyene. Her er et veldig simpelt setup

Det som er rødt betyr relations. Da skaper vi en relasjon mellom de to modellene slik at man kan lett inkludere den modellen mens du henter et verktøy. Da velger man keyen som skal matche i relasjonen.

Prisma funker også med mange type databaser ikke bare postgres.

API endepunktene

**A screenshot of a computer program

AI-generated content may be incorrect.POST /folders**

Først sjekker vi om at brukeren som benytter seg av endpunktet har en valid session igjennom funksjonene modulen nextauth tilbyr. Om de ikke er innlogget sender vi en 401 unauthorized melding som sier at de ikke har tilgang til å utføre handlingen. Om de har en session men brukeren deres ikke er en administrator vil den også si til brukeren at dem ikke har tilgang til å utføre handlingen. Etter alt av autentisering er godkjeft går vi videre til å prosessere formdataen som er sendt inn hvor vi henter ut navn og beskrivelse for mappen. Så sjekker vi med zod om form dataen som er sendt inn er OK. Om den ikke er OK sender vi en form validation failed feilmelding til brukeren.

Så sjekker vi om at mappenavnet er opptatt og sender en feilmelding om det oppstår en konflikt.

Når alt er gjort bruker vi prisma til å opprette mappen i databasen og revaliderer urlen slik at endringene som ble utført blir dynamisk hentet umiddelbart av klientene.

Sender da en sukse melding til brukeren.

**A screen shot of a computer program

AI-generated content may be incorrect.DELETE /folders/[folderId]**

Her gjør vi som alle viktige eksponerte endepunkter, sjekker at dem er innlogget og har type som administrator før vi prosesserer dataen som er innsendt for å minimere stress på database / server.

Når det er gjort henter vi ut hvilken mappe det er snakk om som vi da sjekker om en mappe med den ID’en finnes. Om den ikke finnes returnerer vi en melding til brukeren som sier at vi ikke fant den forespurte mappen.

Vi sletter mappen med bruk av prisma forså å revalidere urlen slik at brukeren henter inn endringene dem selv har utført.

Sender da en sukse melding til brukeren.

**A screenshot of a computer program

AI-generated content may be incorrect.POST /tools**

Her gjør vi som alle viktige eksponerte endepunkter, sjekker at dem er innlogget og har type som administrator før vi prosesserer dataen som er innsendt for å minimere stress på database / server.

Henter ut all data fra formdataen som klienten sende over for å validere det med zod. Om formdataen inneholder et bilde sender vi det til Image klassen som da genererer en UID (Sjekker også at denne uiden ikke er tatt av et annet bilde) forså å converte det til binær og skrive det til /public/uploads (i produksjon ville dette vært en blob storage osv...) Så tar vi vare på imageId og tilknytter den til mappen når vi oppretter den med prisma.

Så revaliderer vi endringene slik at brukeren får se endringene de har utført.

Sender da en sulse melding til brukeren.

A screen shot of a computer program

AI-generated content may be incorrect.**DELETE /folders/[folderId]**

Sjekker tilganger som vanlig på starten av koden. Også sender vi en oppgave til Folders klassen som da håndterer sletting av mappen.

Klassen folders sletter ikke bare mappen men alle verktøy som ligger inni den. Bildene som er lagt ved verktøyene blir også slettet for å ikke beholde på irrelevant data som ikke trengs lengre. Gjør også tjenesten mye mere effektiv og billigere å kjøre.

A screenshot of a computer program

AI-generated content may be incorrect.**DELETE /tools/[toolId]**

Håndterer session og sjekker om brukeren er autentisert. Om brukeren er autentisert sjekker vi om at brukeren er en administrator. Om brukeren er en administrator sjekker vi om verktøyet dem prøver å slette er et som eksisterer. Om det eksisterer sender vi det inn til custom classen vi har laget som håndterer prisma callet og sletter bildet som verktøyet har.

Om alt går fint sender vi en sukse melding til brukeren.

**Classes**

**A screen shot of a computer program

AI-generated content may be incorrect.**Her benytter jeg meg av innebygde funksjonen class i javascript som tillater object oriented programming. Selv om det ikke ble brukt i dette eksemplet så er det hva funksjonene egentlig er laget for.

A screenshot of a computer program

AI-generated content may be incorrect.

Forms

A screen shot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.For forms har jeg valgt en modul som heter useForm siden den lar deg lett legge in zod schemas for form validation. useForm ligger bare på klient nivå så zod schemaet må sjekkes manuelt på backend.

A screen shot of a computer

AI-generated content may be incorrect.Jeg eksporterer en modul i samme directory som modalen som eksponerer forskjellige form komponenter som trengs for å klistre alt sammen.

Her kan man se at jeg importerer gruppen av forskjellige komponenter fra useformen slik at jeg kan bruke den inni modalen som å registrere inputs.

A computer screen shot of a program code

AI-generated content may be incorrect.

Hjelpemidler

Nextauth dokumentasjon:

<https://next-auth.js.org/providers/credentials>

Avfallsportalen.no (Mitt prosjekt hos iris)

<https://github.com/mathiasvoie/avfallsportalen.iris-salten.no>

Egetprosjekt med autentisering

<https://github.com/mathiasvoie/Playground>