**Fagprøve**

Mathias Voie Johansen 26.05.25

**Er det behov for assistanse:**

* Nei, det er ikke **behov** for assistanse.

**Mulige altvernative løsninger:**

* **Node** (Backend) / Javascript (Frontend) / CSS / MySQL
* **NextJs** / Javascript (Frontend) / TailwindCSS / MySQL / Next-Auth / Prisma
* **Node** (Backend) / Python (Backend) / Javascript (Frontend) / MySQL
* **NextJs** / Typescript / TailwindCSS / PostgreSQL / Next-Auth / Prisma

**Den valgte løsningen (nr 4.):**

**NextJS:** Sterkt rammeverk som har alt fra (*App Router,**Parallell Routes,**Avansert Caching**til**Server-side Rendering)*som jeg føler passer perfekt til dette prosjektet siden det er *Henting, Posting, Oppdatering* av data.

**Typescript:** Forsikrer at utvikleren har full kontroll over egen kode med perfekt **IntelliSense** slik at du finner de verdiene du leter etter umiddelbart. Typescript er også **‘Type-Safe’** slik at om noen typer er feil får man opp feilmeldinger under utviklingen istede for under **Run Time.**

**TailwindCSS:** Gjør det enklere å designe nettsiden med innebygde farger og CSS klasser som kan settes på forskjellige HTML elementer. Tailwind bytter ikke ut vanlig CSS men gjør klassene mere tilgjengelig for oss utviklere.

**PostgreSQL:** Open-Source platform som er høyt pålitelig og som er veldig skalerbart for småe og store prosjekter. Følger også SQL standaren som forenkler applikasjonsutvikling og migrering.

**Next-Auth:** Jeg velger dem siden det er hva jeg har brukt igjennom læretiden og at jeg har en forståelse for hvordan ting henger sammen. De har mange integrerte løsninger som man kan bruke for autentisering som f.eks *(Google, GitHub, Credentials osv...)*

**Prisma:** (Object-Relational Mapper)Tilbyr IntelliSense på database queries som er et godt verktøy for utviklere når dem skal hente data fra databasen. Prisma forsikrer også at utvikleren er oppmerksom på feiler og mangler under utvikling istede for i run time.

**Oppgave:**

Oppgaven jeg har vil utføre er en ‘Vektøykasse’ / plass hvor man kan legge inn verktøy i forskjellige mapper som kan kontrolleres / endres av klienter som er innlogget på siden. Bilde opplastning vil jeg skal være en mulighet for en tettere beskrivelse av verktøyene som legges inn. Rolle basert styring vil jeg skal være en funksjon slik at man kan ha ‘Administratorer’ og ‘Brukere’ med hvær sin funksjonalitet.

**Tidsplan:**

**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
* Autentisering
* Sikkerhet for brukere
* Sette av tid til egenvurdering og dokumentasjon

**Onsdag:**

* API ( PUT, PATCH, DELETE )
* 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
* Bildeopplastning til server
* 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