**Projektbeskrivelse for ClimaSense**

**Formål og funktionalitet:** ClimaSense er et projekt, der henter og viser vejrinformationer. På fredag vil appen kunne hente data fra DMI’s API, som indeholder oplysninger som temperatur, tryk, vindhastighed og andre relevante vejrmålinger. Jeg har tænkt at appen skal vise disse data på en enkel og overskuelig måde, så brugeren hurtigt kan få indsigt i vejret for specifikke steder og tidspunkter.

Jeg har valgt at bygge appen med React Native, så den både kan køre på iOS og Android. Det betyder, at jeg kan udvikle én version af app’n, som virker på begge platforme. Appen henter data fra min backend, som henter det fra DMI, og viser dem i en simpel og brugervenlig form.

**Funktioner på fredag:**

* Hente vejrinformationer fra DMI’s API via backend.
* Vise data som f.eks. temperatur, vindhastighed og lufttryk og andre ting.
* Dataene præsenteres i "cards", hvor hver kort viser parameteren, dens værdi og tidsstempler.
* En simpel header for at vise appens navn og funktion.

**Teknologier:**

* **React Native:** Jeg bruger React Native, fordi det giver mig mulighed for at bygge én app, der virker på både iOS og Android. Det sparer tid på udvikling og vedligeholdelse, da jeg kun behøver én kodebase for begge platforme. Det er også en meget populær teknologi, der er let at arbejde med.
* **Axios:** Til at hente data fra backend bruger jeg Axios, da det er et effektivt og simpelt bibliotek til HTTP-anmodninger. Jeg skal bruge Axios til at hente vejrinformationerne fra DMI’s API og sende dem videre til frontend.
* **Express og PostgreSQL:** Backend er bygget med Express, da det er det der hedder et letvægts-webframework for Node.js, som gør det hurtigt at udvikle et API. Jeg bruger PostgreSQL som database, fordi det er en pålidelig og skalerbar løsning til at gemme de vejrinformationer, man f.eks favoritter. Det er godt til at håndtere strukturerede data og giver mig stor fleksibilitet, når jeg arbejder med forespørgsler og opbevaring af data.

**Begrundelse for valg af teknologier:**

* **React Native:** Jeg valgte React Native, fordi det gør det muligt at udvikle én applikation til både iOS og Android. Jeg slipper for at udvikle to separate apps og kan spare tid, samtidig med at jeg får en god brugeroplevelse på begge platforme.
* **Axios og Express:** Jeg bruger Axios, fordi det er nemt at integrere med React Native og gør HTTP-anmodninger hurtigt og effektivt. Express er et ideelt valg for backend, da det er enkelt at arbejde med og hurtigt kan skaleres.
* **PostgreSQL:** PostgreSQL giver mig en stabil og effektiv database til at håndtere de vejrinformationer, vi arbejder med. Det er en relationsdatabase, der er god til at organisere og gemme store mængder strukturerede data.