**Afsluttende Projektrapport – ClimaSense**

**Indledning**

ClimaSense er et projekt, hvor jeg skulle hente og vise vejrinformationer fra DMI’s API og præsentere dem på en brugervenlig måde i en mobilapp. Målet var at skabe en simpel app, hvor brugeren hurtigt kunne få overblik over temperatur, vindhastighed, lufttryk og andre relevante data. Appen blev udviklet i **React Native**, så den kunne køre på både iOS og Android. Backend blev lavet med **Express**, og jeg planlagde at bruge **PostgreSQL** til at gemme vejrinformationer, f.eks. favoritter. Du kan se screenshots af appen nederst på sidenz.

Undervejs stødte jeg dog på flere udfordringer, som betød, at ikke alt blev implementeret som oprindeligt planlagt.

**Hvad jeg nåede, og hvad jeg ikke nåede**

**Hvad jeg fik implementeret:**

* Hente vejrinformationer fra DMI til min egen backend
* Vise temperatur, vindhastighed og tidspunkter i en overskuelig brugerflade
* Bygge en simpel app med React Native og få den til at hente data fra en ekstern kilde
* Formatere data og præsentere dem pænt i "cards"

**Hvad jeg ikke nåede, og hvorfor:**

* **Brug af DMI’s oprindelige API** – Jeg startede med at prøve at bruge DMI’s API, som leverer data i **GRIB-filer**. Disse filer er i et specielt binært format, der kræver ekstra konvertering for at kunne læses og bruges i appen. Det viste sig at være meget besværligt at arbejde med, da jeg skulle bruge en specifik parser til at dekode filerne og ekstrahere de relevante data. Efter flere forsøg, hvor jeg stødte på problemer med både formateringen og tilgængeligheden af dataene, valgte jeg at skifte til et andet API-endpoint fra DMI, som leverede data i JSON-format. Dette gjorde det meget lettere at hente og vise informationerne i appen.
* **Implementering af PostgreSQL-database** – Planen var at gemme brugerens favoritter i en database, men pga. tidsmangel og andre tekniske problemer nåede jeg ikke at sætte en fuld backend op til dette. Jeg fokuserede i stedet på at få appens grundlæggende funktioner til at fungere.
* **Flere vejrparametre** – Jeg ville oprindeligt vise flere målinger som lufttryk, luftfugtighed osv., men da jeg prioriterede at få basale funktioner op at køre, valgte jeg at holde mig til temperatur, vindhastighed og tid.

**Udfordringer undervejs**

1. **Problemer med API’er** – Jeg havde regnet med at bruge DMI’s API, men det viste sig at være sværere at arbejde med end forventet. Oprindeligt arbejdede jeg med GRIB-filer, men jeg fandt hurtigt ud af, at de var komplicerede at parse og krævede ekstra behandling. Derfor skiftede jeg til et andet DMI-endpoint, der returnerede data i JSON-format, hvilket var meget lettere at håndtere.
2. **Databaseopsætning tog længere tid end forventet** – Jeg havde planlagt at bruge PostgreSQL til at gemme favoritter, men jeg brugte meget tid på at få frontend og backend til at snakke sammen, så database-delen blev skubbet til sidst og nåede ikke at blive implementeret.
3. **React Native var nyt for mig** – Da jeg ikke havde arbejdet med React Native før, brugte jeg tid på at lære frameworket at kende. Det tog lidt ekstra tid i starten, men jeg fik hurtigt styr på det grundlæggende.
4. **Tidsmangel** – Jeg havde kun en begrænset tidsramme til projektet, og jeg måtte derfor prioritere de vigtigste funktioner og skære visse ting fra.

**Hvad jeg har lært**

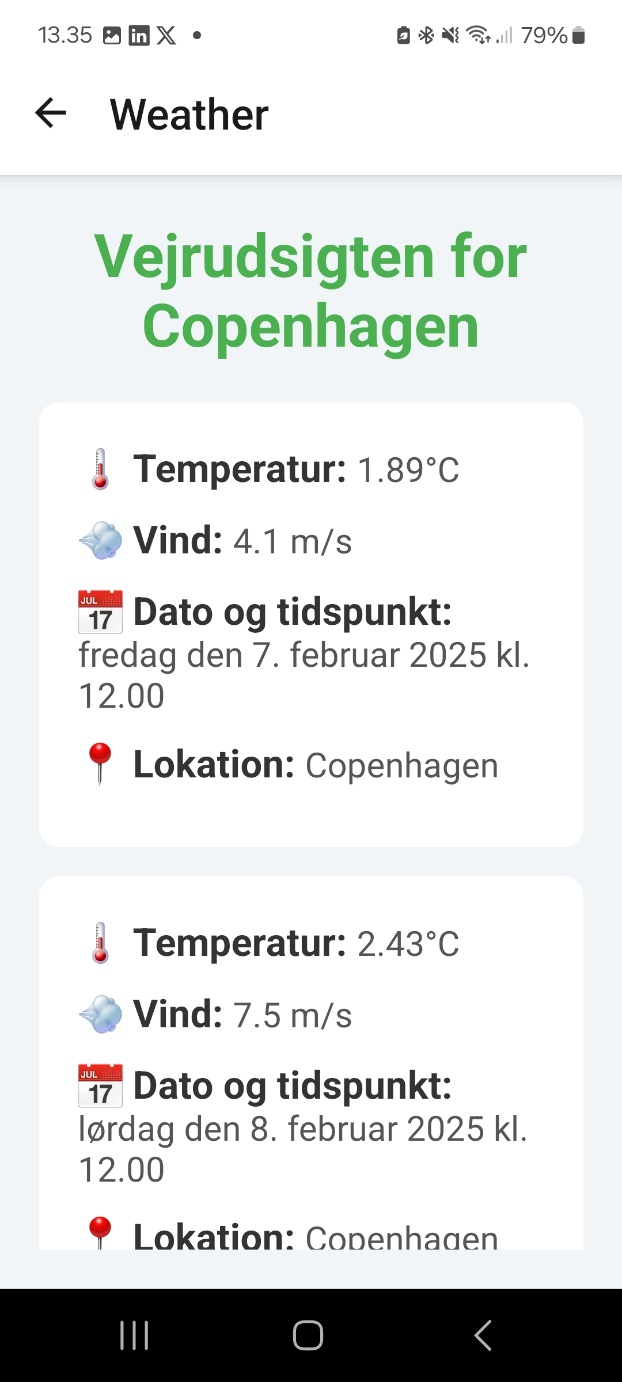
* **Bedre planlægning af API-integration** – Jeg har lært, at det er vigtigt at teste API’er grundigt, før jeg baserer et helt projekt på dem. Hvis jeg havde opdaget DMI’s API-begrænsninger tidligere, kunne jeg måske have valgt en anden løsning fra starten.
* **React Native er spændende, men kræver tid at lære** – Selvom det var nyt for mig, har jeg fået en bedre forståelse af, hvordan man bygger mobilapps, og hvordan React Native fungerer.
* **Backend og frontend skal testes sammen tidligt** – Jeg brugte en del tid på at udvikle backend og frontend hver for sig, men jeg kunne have testet dem sammen tidligere for at undgå integrationproblemer senere.
* **Tidsstyring er vigtig** – Jeg kunne måske have prioriteret anderledes, så jeg nåede flere af de planlagte funktioner. Næste gang kan jeg lave en mere realistisk plan og afsætte mere tid til fejlretning.

**Konklusion**

Selvom jeg ikke fik implementeret alt, hvad jeg oprindeligt havde planlagt, er jeg stadig tilfreds med det, jeg har opnået. Jeg har lært en masse om React Native, API-integration og backend-udvikling, og jeg har fået en fungerende app, der kan vise vejrinformationer på en brugervenlig måde. Hvis jeg skulle fortsætte projektet, ville jeg arbejde videre med at integrere en stabil vejrdatakilde og tilføje databasefunktionalitet, så brugeren kan gemme favoritter.

Overordnet set har projektet givet mig værdifuld erfaring, som jeg kan bruge i fremtidige udviklingsopgaver.

**Screenshots**

** **