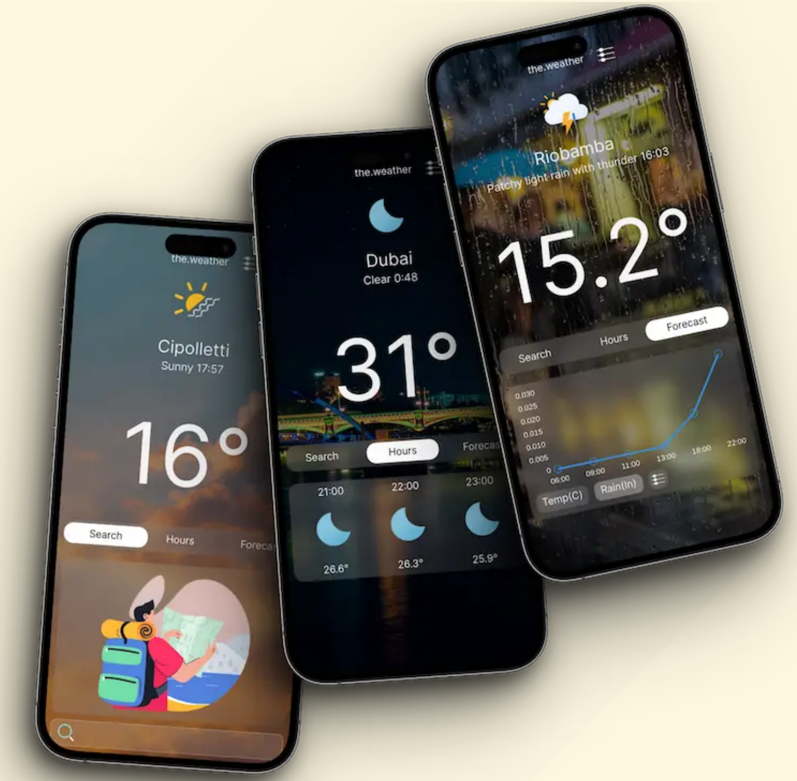
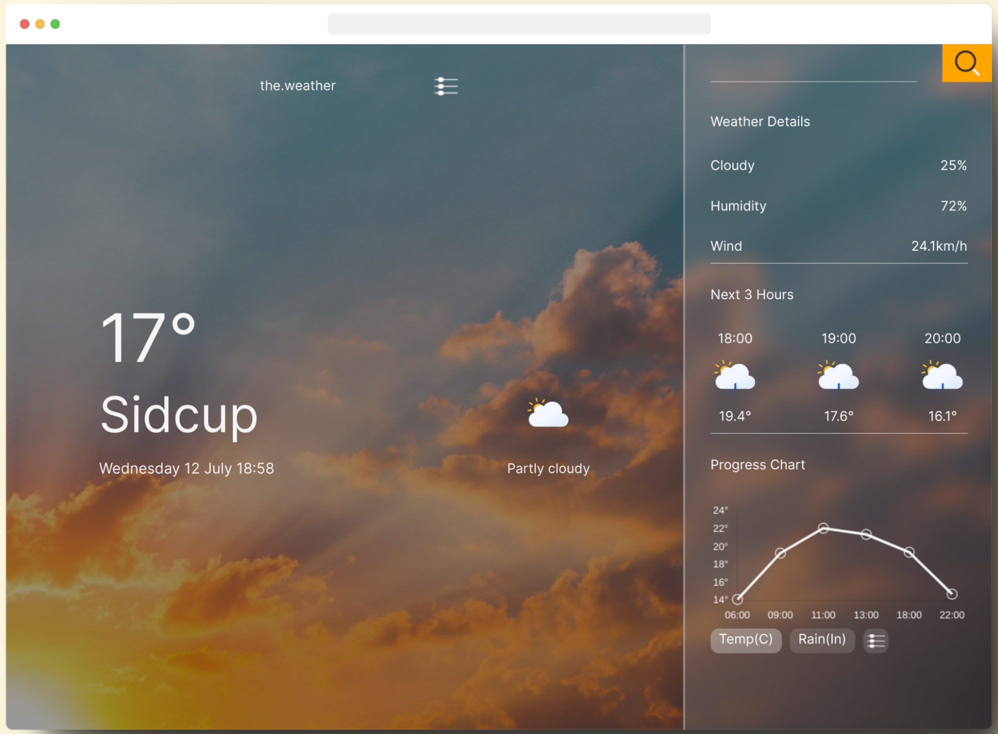
Weather project overview

By Samuel Ademola





Weather Details

Cloudy	25%
Humidity	72%
Wind	24.1km/h

Next 3 Hours

18:00	19:00	20:00
19.4°	17.6°	16.1°

Contents

Project

A weather PWA

Project features

- PWA
- Page themeing based on data
- Adaptive design
- interacts with an API
- City based search functionality

Project focus

- Theme redesign/New theme
    - Most of the code, I had built previously so this was more of a UX project
- Improve perfomance and stability
- Implement Search

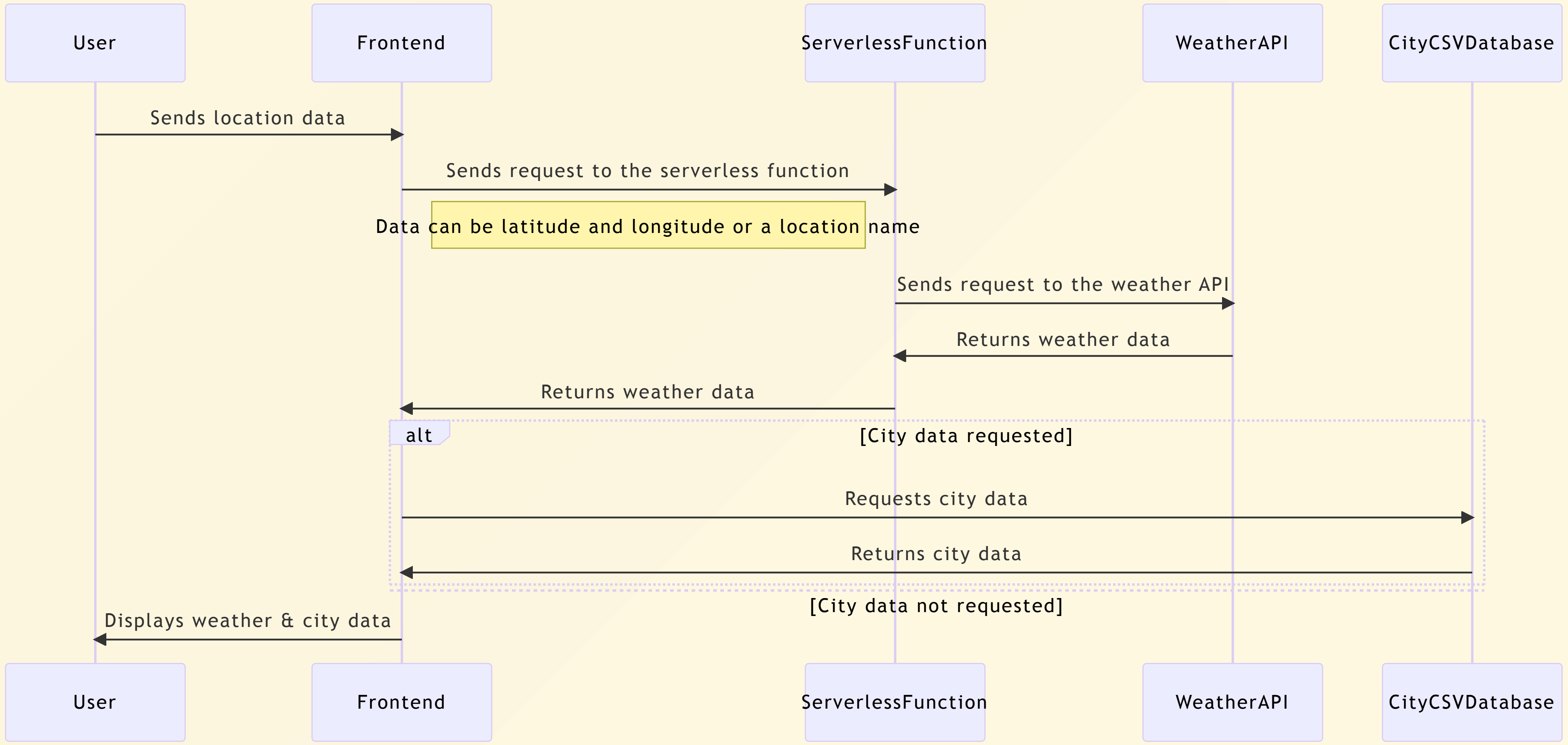
Demonstaration

Project structure

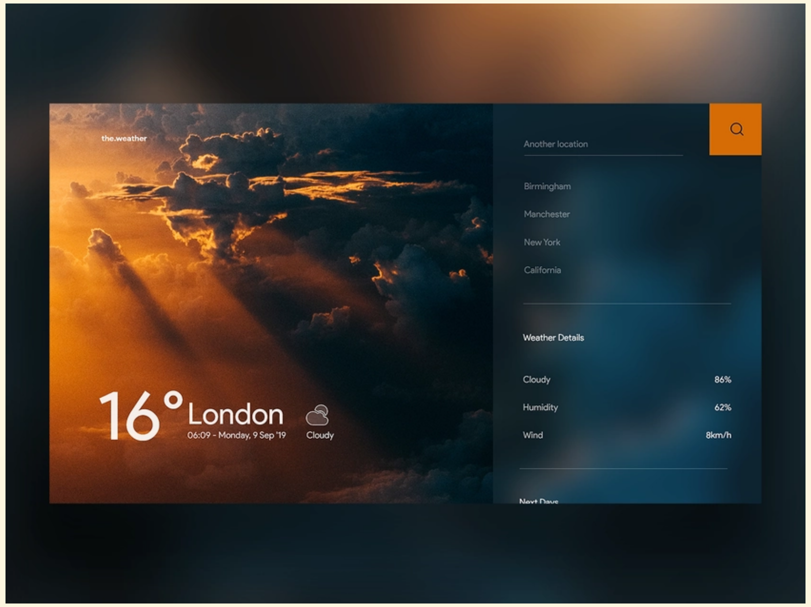
```
├── api
├── dist
├── node_modules
├── public
├── src
│   ├── api
│   ├── assets
│   ├── components
│   ├── composables
│   ├── router
│   ├── stylesheets
│   ├── views
│   └── webworkers
└── useful-notes-here
```

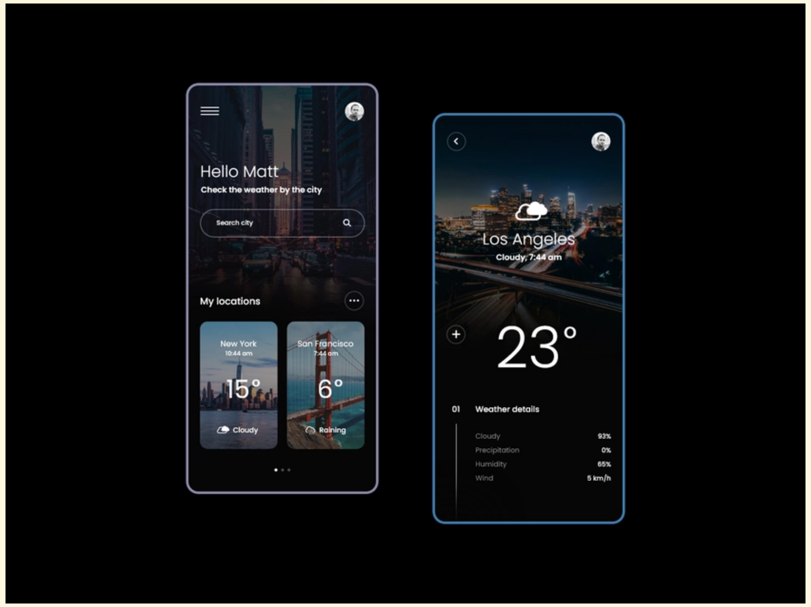
Project Stack

- Built on Vue
- Site and assets are served through Vercel
- Serveless function also through vercel
- Api from https://www.weatherapi.com/


User
Frontend
ServerlessFunction
WeatherAPI
CityCSVDatabase
Sends location data
Sends request to the serverless function
Data can be latitude and longitude or a location name
Sends request to the weather API
Returns weather data
Returns weather data
alt
[City data requested]
Requests city data
Returns city data
[City data not requested]
Displays weather & city data
User
Frontend
ServerlessFunction
WeatherAPI
CityCSVDatabase

Design Inspo





https://dribbble.com/sam4356/collections/6353619-Weather-app

Interesting features

Search

- Fast (So want to avoid API)
  - lightweight solution(if local)
- Extensive search
- Asynchronous
  - non ui blocking

solution

Web worker with a csv databas in the front end

```
// Listen for messages from the main thread
self.onmessage = (event) => {
  const {searchTerm } = event.data;
  // console.log("csv", searchTerm)

  fetch("./../assets/cities-new.csv")
    .then((response) => response.text())
    .then((csvString) => {
      // Parse the CSV file into an array of objects
      const csvData = Papa.parse(csvString, { header: true }).data;

      // Create a Fuse.js instance with the options you want
      const fuse = new Fuse(csvData, {
        keys: ["name"], // Search on the 'name' and 'email' fields
        includeScore: true, // Include the score in the results
      });
      // Search for the searchTerm in the CSV data
      const results = fuse.search(searchTerm);

      // Trim the results to the first 4 matching objects
      const trimmedResults = results.slice(0, 4);

      // Send the results back to the main thread
      self.postMessage(trimmedResults);
      close();
    })
    .catch((error) => {
      // Send an error message back to the main thread
      console.log({ error: error.message })
      self.postMessage({ error: error.message });
    });
};
// Export the worker script as a default export
export default "searchCSV.js";
```

Data based styling/theming

- LUT
- the api provides conditions code/names for day/night
- based on the object in the lUT weather data we get the svg which best represents the condition and then apply a class to the svg

```
{
        "code": 1003,
        "day": "Partly cloudy",
        "night": "Partly cloudy",
        "icon": 116,
        "day-img": "partly-cloudy-day.svg",
        "night-img": "partly-cloudy-night.svg",
        "night-css": "cloudy",
        "day-css": "cloudy"
},
```

```
    let svgName = conditionIsDay ? weatherLUT.find(({ day }) => day == props.condition) : weatherLUT.find(({ night }) => night == props.condition)
...

      svgSrc.value = require(`@/assets/svg/${svgName[conditionIsDay ? 'day-img' : 'night-img']}`);
```

Data based styling/theme

```
:root:has(img.icon.snow) {
  @media (min-width: 0px) {
    // and at 500px and 1200px
    div.outer-bg {
      background: url("./../assets/theme-new/bg-1200.webp") center / cover
        no-repeat;
    }
  }
}
```

Page load performance in my weather SPA

My control for page performance - google lighthouse

FCP - First Contentful Paint	TBT - Total Blocking Time	LCP - largest Contentful Paint
5.5s -> 1.9s	0ms - 110ms	5.5s -> 3.9s

Page load performance in my weather SPA(contd)

Suggestions by lighthouse

Eliminate render-blocking resources -
main css file - 1.31s
reduce unused javascript - 1.2s

- lottie js 90.3/335 kbs
- vuesax.common.js 66kbs
- gsap-core.js 15.3kbs

steps to fix this

- I created my own vue js Lottie player component and used a lazy-loaded script tag
- recreated the CSS I used Vuesax common for and created my own slider

- recreated my vue trasnitions using css only and removed gsap