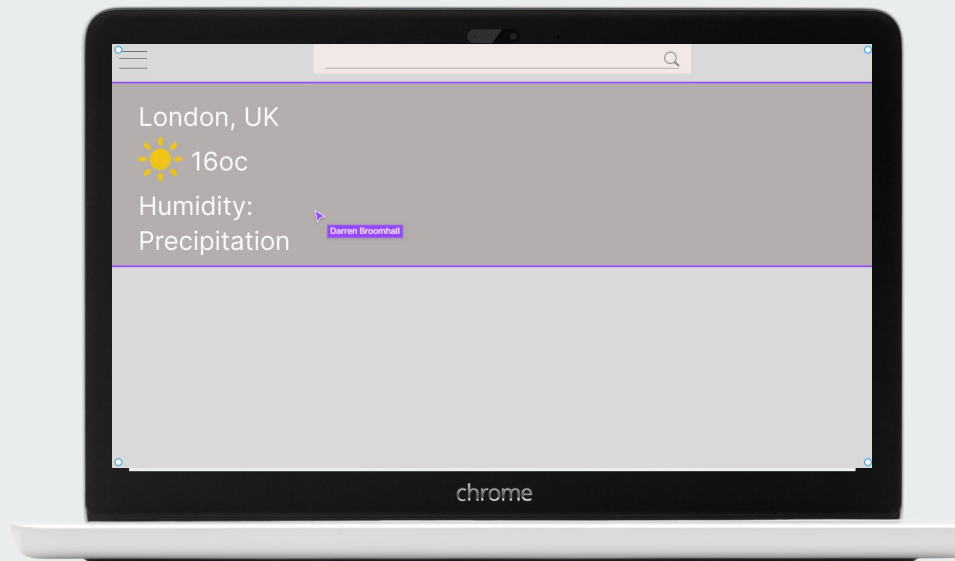




Typescript- React Weather App

Bidhan, Catherine, Darren, Naomi
Hackathon - Week 11



Planning

The screenshot shows a Trello board titled "Typescript Reactathon" with a yellow background and a pattern of orange and yellow spheres. The board is organized into five columns: Resources, To do, Doing, Done, and Timetable. The "Resources" column contains links to GitHub repositories and articles, as well as API references for Weather and Geocoding. The "To do" column lists tasks like "Plan extra feature", "Optional bonus tasks", "CSS Styling", "Animations", "Tests", and "Code app". The "Doing" column includes "Plan timetable for day", "Component Tree", "Organise API calls", "Plan API use", and "Wireframe". The "Done" column has "Sign up for free tier on API". The "Timetable" column shows a schedule from 9:30 to 12:30, including planning tasks and coding time. The Trello interface includes a top navigation bar with "Workspaces", "Recent", "Starred", "Templates", and "Create" buttons, a search bar, and a bottom bar with "Power-Ups", "Automation", "Filter", and "Share" options.

Trello Board: Typescript Reactathon

Resources

- REPO : https://github.com/SchoolOfCode/bc14_w11d5_hackathon_react-typescript-room_21_darren_naomi_catherine
- links: <https://github.com/typescript-cheatsheets/react#react-typescript-cheatsheets>
<https://profy.dev/article/react-typescript>
<https://www.carlrippon.com/React-event-handlers-with-typescript/>
- 2
- Weather API**
Simple and fast and free weather API from OpenWeatherMap you have access to...
- Openweathermap
- Geocoding API**

To do

- Plan extra feature
- Optional bonus tasks
- CSS Styling
- Animations
- Tests
- Code app
- + Add a card

Doing

- Plan timetable for day
- Component Tree
- Organise API calls
- Plan API use
- Wireframe
- + Add a card

Done

- Sign up for free tier on API
- + Add a card

Timetable

- 9.30 - 10: Plan: Trello, components, extra feature?
- 10-11.30: Create component tree: state, functions, rendering. Organise folder structure in app.
- 11.30 - 12: Begin coding the app.
- 12 - 12.30: Look over progress of the morning. Plan the afternoon.
- + Add a card

TWO API CALLS:

1. FETCH LAT AND LONG from geogoding API using city name
2. use lat and long to fetch WEATHER from weather aPI onecall 3.0

Naomi Crisp

```
},  
  "lat":51.5073219,  
  "lon":-0.1276474,  
  "country":"GB",  
  "state":"England"  
},
```

```
async function fetchLatLonAPI(search) {  
  const url1 = 'http://api.openweathermap.org/  
geo/1.0/direct?q=${search}&units=metric  
&limit=1&appid=03eeaa1ba18c8cd073330aef774  
df3e8'  
  const response1 = await fetch(url1)  
  const data1 = await response1.json()  
  const lat = data1.lat  
  const lon = data1.lon  
  const url2 = 'https://api.openweathermap.org/  
data/3.0/onecall?lat=${lat}&lon=${lon}  
&appid=03eeaa1ba18c8cd073330aef774df3e8'  
  const response2 = await fetch(url2)  
  const data2 = await response2.json()  
  return data2  
}
```

STRETCH GOALS

Make location widgets that you can click on to bring up a more detailed 24hr weather report timeline. Have toggles for hourly and daily above the timeline.

Darren Broomhall

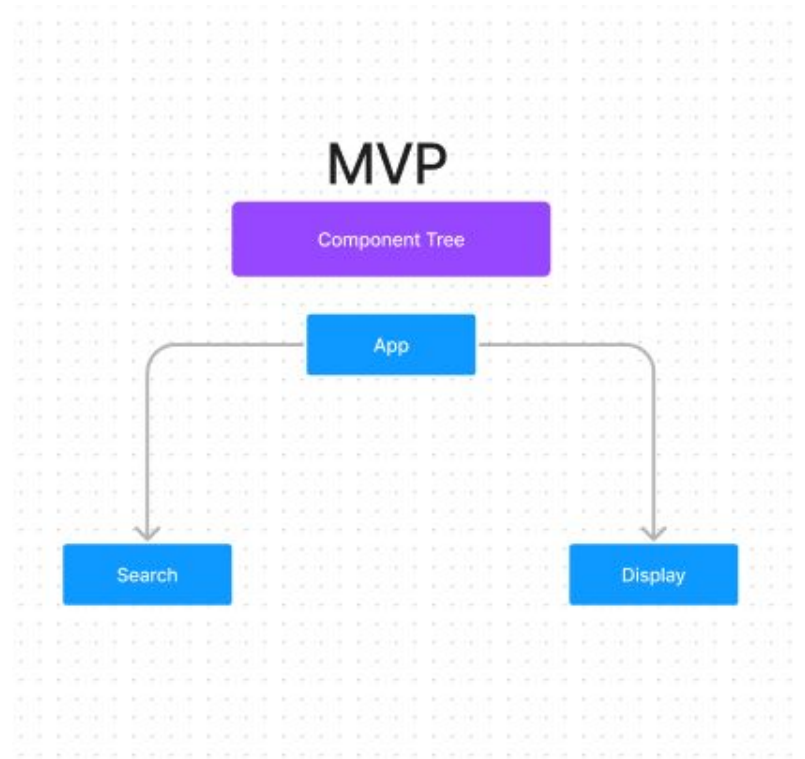
Make a menu/filter to more easily bring up a record for weather hourly/ daily - Bidhan

Darren Broomhall

******CORRECTION: ONLY ONE API CALL WAS REQUIRED.**

PSA: CHECK EMAILS.

Component tree



MVP

Component Tree

App

```
type WeatherState = {...}  
const [weather, setWeather] = useState<WeatherState>({})
```

1. fetch API and save to state

Search

```
<input></input>  
<button></button>
```

```
const [location, setLocation] = useState<string>({})
```

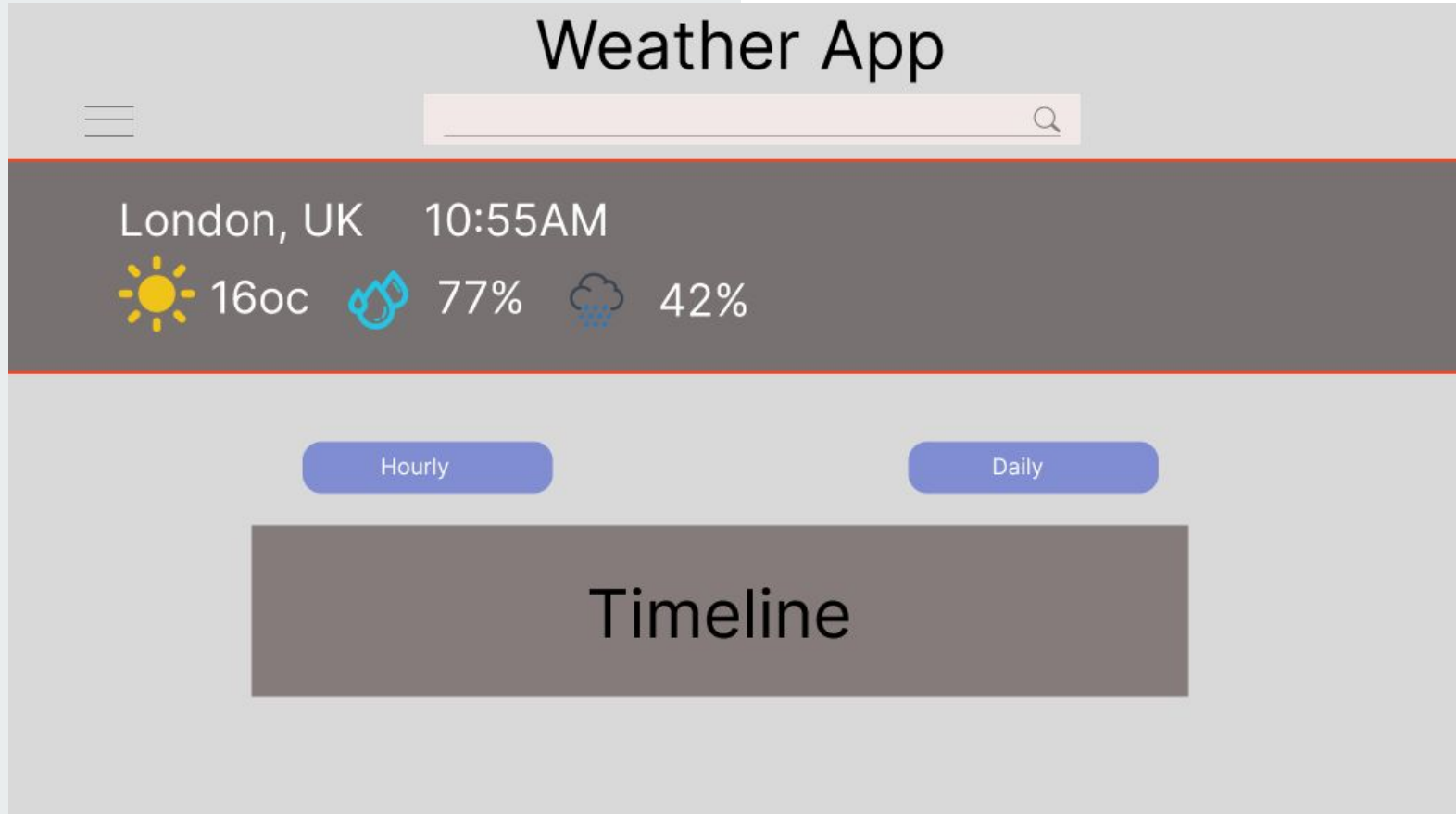
1. save the input into state
2. Call the fetch function on button click with search state.

Display

```
<section>  
  <h1 id="location"></h1>  
  <h2 id="current-temp"></h2>  
  <img ???/>  
  <h4></h4>  
  <h4></h4>  
  <h4></h4>  
  <h3>TIME</h3>  
</section>
```

Component tree + code planning

Wireframe



The real thing:

