

public/index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
  <link href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/@mdi/font@4.x/css/materialdesignicons.min.css" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/vuetify@2.x/dist/vuetify.min.css" rel="stylesheet">
  <title>5 Day Weather</title>
</head>

<body>
  <div id="app">
    <v-app id="inspire">
      <div style="display: flex; justify-content: center; margin: 0 10px; align-items: center;">

        <v-col cols="3" sm="3" class="mt-8">
          <v-text-field v-model="locationInput" label="Location" clearable outlined dense></v-text-field>
        </v-col>
        <v-btn color="primary" class="mt-1" @click="getWeather" depressed>
          <v-icon left>
            mdi-cloud-search-outline
          </v-icon>
          Search
        </v-btn>
        <span class="red--text font-weight-bold ml-5" style="width: 20%;">
          {{errorMessage}}</span>
        </div>

        <v-main>
          <div v-if="weatherData.length≠0" style="display: flex; flex-direction: column; align-items: center; font-size: large; font-weight: bold;">

            <div v-if="willRain">
              You will need to bring an umbrella today!
            </div>
            <div>
              {{weatherString}}
            </div>
          </div>
          <v-container>
            <div style="display: flex; flex-direction: column;">
              <template v-for="(value,index) of weatherData">
                <div>
                  <span class="text-h4 font-weight-light">{{ value[0].timeString.split(" ")[0] }}</span>
                </div>
                <div style="display: flex; flex-direction: row; gap: 10px;">
                  <div v-for="(weatherInstance, weatherIndex) of value" :key="`${index}${weatherIndex}`" style="width: fit-content;">
                    <v-sheet>
```

```

decoration: underline;" class="">>
)[1].substring(0, 5)}}

<div style="font-size: 1.4rem; text-
    {{value[weatherIndex].timeString.split(" "
</div>
<div class="">>
    {{value[weatherIndex].temp}}°C
</div>
<div class="">>
    {{value[weatherIndex].rain}}mm Rain
</div>
<div class="">>
    {{value[weatherIndex].wind.speed}}m/s Wind
</div>
</v-sheet>
</div>
</div>
</template>
</div>
</v-container>
</v-main>
</div>

<script src="https://cdn.jsdelivr.net/npm/vue@2.x/dist/vue.js"></script>
<script src="https://cdn.jsdelivr.net/npm/vuetify@2.x/dist/vuetify.js"></script>
<script >
    const BASE_URL = "http://localhost:5501/"
    const UMBRELLA_THRESHOLD = 2; //min amount of rain required for umbrella (in
mm)
    const COLD_THRESHOLD = 10; //any temp below this is cold
    const HOT_THRESHOLD = 20; //any temp above this is hot
    new Vue({
        el: '#app',
        data: {
            "locationInput": "Dublin,IE",
            "drawer": true,
            "weatherData": [],
            "willRain": false,
            "coldWeather": false,
            "warmWeather": false,
            "hotWeather": false,
            "errorMessage": "",
            "airQual": ""
        },
        computed: {
            weatherString: function () {
                let weatherArr = [];
                if (this.coldWeather)
                    weatherArr.push("cold");
                if (this.warmWeather)
                    weatherArr.push("warm");
                if (this.hotWeather)
                    weatherArr.push("hot");
                return weatherArr.length > 0 ? `It will be ${weatherArr.join(", ")}
} today` : "";
            },
        },
        methods: {
            getWeather: function () {
                axios.get(BASE_URL + "5dayweather", {
                    params: {

```

```

        location: this.locationInput
      }
    })
    .then(res => res.data)
    .then(res => {
      if (res.error) { //handle API errors
        this.errorMessage = res.error;
        return
      }
      this.errorMessage = ""
      let weather = res.weather;
      weather.forEach(element => { //determine what to inform
        // only care about the current day
        let dataDate = new Date(element.timeString)
        if(dataDate.getDay() == new Date().getDay()) {
          console.log(dataDate)
          this.willRain = this.willRain || element.rain >
            UMBRELLA_THRESHOLD;
          this.coldWeather = this.coldWeather ||
            element.temp < COLD_THRESHOLD;
          this.warmWeather = this.warmWeather ||
            (element.temp > COLD_THRESHOLD && element.temp < HOT_THRESHOLD);
          this.hotWeather = this.hotWeather || element.temp
            > HOT_THRESHOLD;
        }
      });
      this.weatherData = sortToArrayOfDays(weather)
      axios.get(BASE_URL+"airqual", {
        params: {
          lat: res.city.coord.lat,
          lon: res.city.coord.lon
        }
      }).then(res => res.data)
      .then(res => {
        if (res.error) {
          this.errorMessage = `failed to get airqual:
            ${res.error}`
          return
        }
        this.errorMessage = ""
        console.log(res)
        airData = res.airData;
        this.airQual = ""
        airData.forEach(element => {
          if (element.aqi > 10) {
            this.airQual = "Air Quality is bad, wear a
              mask"
          }
        })
      })
    })
    console.log(this.weatherData)
  })
}

},
vuetify: new Vuetify(),
})
//reorganise the data to make it easier to display
function sortToArrayOfDays(inputArray) {
  let returnArray = [];
  let curIndex = -1;
  let curDate = "";

```

```
    for (element of inputArray) {
      let dayStr = element.timeString.split(" ")[0];
      // console.log(dayStr)
      if (dayStr !== curDate) {
        curDate = dayStr;
        curIndex++;
        returnArray[curIndex] = [];
      }
      returnArray[curIndex].push(element)
    }
    return returnArray;
  }
</script>
</body>

</html>
```