LOVELY PROFESSIONAL UNIVERSITY

NAME: TARUN

REG. NO.:12110483

SUBJECT CODE: INT303

SECTION: K21PK

ROLL NO.: RK21PKB49

PROJECT NAME: WEATHER WEB APPLICATION

SUBMITTED TO:

NAHIDA NAZIR MA'AM

CONTENTS

ACKNOWLEDGEMENT

It is great happiness and privilege for me to represent this Project report. I have completed the development of WEATHER WEBSITE as project under the supervision of NAHIDA MAM.

I would like to express my gratitude towards all those people who have in various ways, helped me in successful completion of my project.

I'd like to be thankful to my colleagues and team members for their valuable support and corporation during my project.

Abstract

Weather forecasting is the prediction of the state of the atmosphere for a given location using the application of science and technology. This includes temperature, rain, cloudiness, wind speed, and humidity. Weather warnings are a special kind of short-range forecast carried out for the protection of human life. Weather warnings are issued by the governments throughout the world for all kinds of threatening weather events including tropical storms and tropical cyclones depending upon the location. The forecast may be short-range or Long-range. It is a very interesting and challenging task. This report provides a basic understanding of the purpose and scope of weather forecasts, the

basic principles and the general models developed for forecasting.

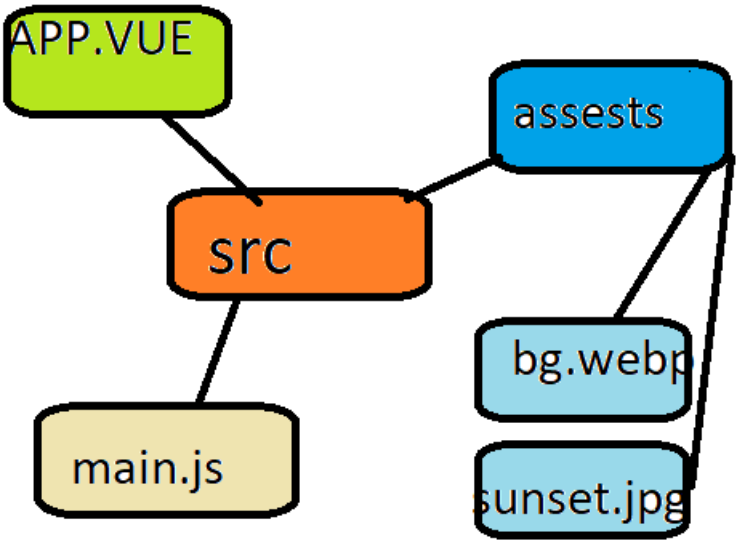
INTRODUCTION

Weather forecasting is the prediction of the state of the atmosphere for a given location using

the application of science and technology. This includes temperature, rain, cloudiness, wind speed, and humidity. Weather warnings are a special kind of short-range forecast carried out for the protection of human life.

This a single page web application made by me for our college project for an weather based temperature and atmosphere calling website with the help of different API server’s.

FLOW CHART



CODE FOR MY WEB APPLICATION

APP.VUE FILE CODE::

```
<template>
  <div
    id="app"
    :class="
      typeof weather.main != 'undefined' && weather.main.temp > 16 ? 'warm' :
''
    "
  >
    <main>
      <center>
        <U><H1>WEATHER WORLD</H1></U>
      </center>
      <br />

      <div class="search-box">
        <input
          type="text"
```

```
          class="search-bar"
          placeholder="Search..."
          v-model="query"
          @keypress="fetchWeather"
        />
      </div>
    <hr>
    <br>

      <div class="weather-wrap" v-if="typeof weather.main != 'undefined'
      ">
        <div class="location-box">
          <div class="location">
            {{ weather.name }}, {{ weather.sys.country }}
          </div>
          <br>
          <B>
            <hr style="color:aquamarine">
          </B>

          <br>
          <center><div><h1 style ="color: white;">{{ timestamp
}}</h1></div></center>
          <br>

          <div class="date">{{ dateBuilder() }}</div>
        </div>

        <div class="weather-box">
          <div class="temp">{{ Math.round(weather.main.temp) }}°c</div>
          <div class="weather">{{ weather.weather[0].main }}</div>
        </div>
      </div>
    </main>
  </div>
</template>

<script>


export default {
  name: "app",
  data() {
    return {
      api_key: "ff0148743305d28f98a6e79c11ea1c02",
      url_base: "https://api.openweathermap.org/data/2.5/",
      query: "",
      weather: {},
      timestamp:"",
```

```
    };
  },
  created() {
                setInterval(this.getNow, 1000);
            },
  methods: {
    fetchWeather(e) {
      if (e.key == "Enter") {
        fetch(
          `${this.url_base}weather?q=${this.query}&units=metric&APPID=${this.a
pi_key}`
        )
          .then((res) => {
            return res.json();
          })
          .then(this.setResults);
      }
    },
    setResults(results) {
      this.weather = results;
    },
    dateBuilder() {
      let d = new Date();
      let months = [
        "January",
        "February",
        "March",
        "April",
        "May",
        "June",
        "July",
        "August",
        "September",
        "October",
        "November",
        "December",
      ];
      let days = [
        "Sunday",
        "Monday",
        "Tuesday",
        "Wednesday",
        "Thursday",
        "Friday",
        "Saturday",
      ];
      let day = days[d.getDay()];
      let date = d.getDate();
```

```
      let month = months[d.getMonth()];
      let year = d.getFullYear();

      return `${day} ${date} ${month} ${year}`;
    },
    getNow: function() {
                    const today = new Date();
                    const time ="Cureent time :"+ today.getHours() + ":" +
today.getMinutes() + ":" + today.getSeconds();
                    this.timestamp = time;
  },
},
}
</script>

<style>
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
body {
  font-family: "montserrat", sans-serif;
}
#app {
  background-image: url("./assets/bg.webp");
  background-size: cover;
  background-position: bottom;
  transition: 0.4s;
}
#app.warm {
  background-image: url("./assets/sunset.jpg");
}

main {
  min-height: 100vh;
  padding: 25px;
  background-image: linear-gradient(
    to bottom,
    rgba(0, 0, 0, 0.25),
    rgba(0, 0, 0, 0.75)
  );
}
.search-box {
  width: 100%;
  margin-bottom: 30px;
}
.search-box .search-bar {
```

```
  display: block;
  width: 100%;
  padding: 15px;

  color: #101010;
  font-size: 20px;
  appearance: none;
  border: none;
  outline: none;
  background: none;
  box-shadow: 0px 0px 8px rgba(0, 0, 0, 0.25);
  background-color: rgba(255, 255, 255, 0.5);
  border-radius: 0px 16px 0px 16px;
  transition: 0.4s;
  background-color: rgb(252, 230, 204);
}
.search-box .search-bar:focus {
  box-shadow: 0px 0px 16px rgba(0, 0, 0, 0.25);
  background-color: rgba(255, 255, 255, 0.75);
  border-radius: 16px 0px 16px 0px;
  background-color: rgb(167, 231, 255);
}
.location-box .location {
  color: #fff;
  font-size: 32px;
  font-weight: 500;
  text-align: center;
  text-shadow: 1px 3px rgba(0, 0, 0, 0.25);
}
.location-box .date {
  color: #fff;
  font-size: 20px;
  font-weight: 300;
  font-style: italic;
  text-align: center;
}

.weather-box {
  text-align: center;
}
.weather-box .temp {
  display: inline-block;
  padding: 10px 25px;
  color: #fff;
  font-size: 102px;
  font-weight: 900;
  text-shadow: 3px 6px rgba(0, 0, 0, 0.25);
  background-color: rgba(255, 255, 255, 0.25);
```

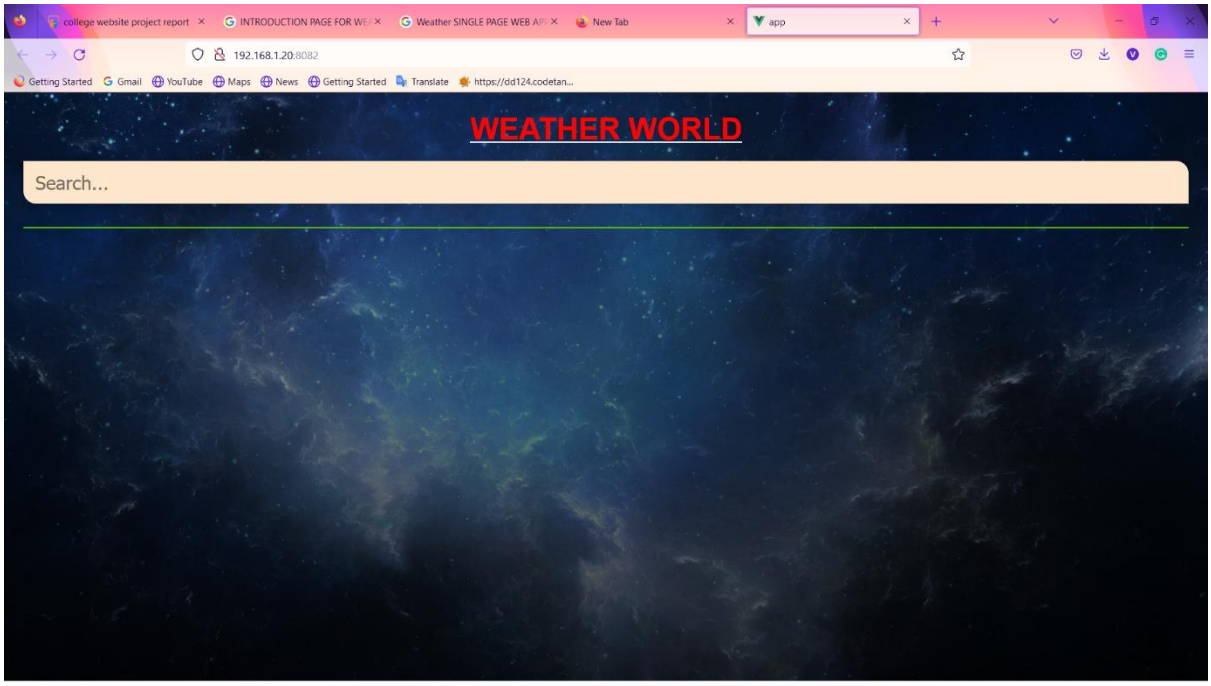
```
  border-radius: 16px;
  margin: 30px 0px;
  box-shadow: 3px 6px rgba(0, 0, 0, 0.25);
}
.weather-box .weather {
  color: #fff;
  font-size: 48px;
  font-weight: 700;
  font-style: italic;
  text-shadow: 3px 6px rgba(0, 0, 0, 0.25);
}
h1 {
  color: rgb(250, 4, 4);
}
u {
  color: white;
  font-size: larger;
}
::placeholder{
  font-size: larger;
  color:rgb(0, 0, 0);
}
hr{
  color: chartreuse;
}
</style>
```

main.js-:

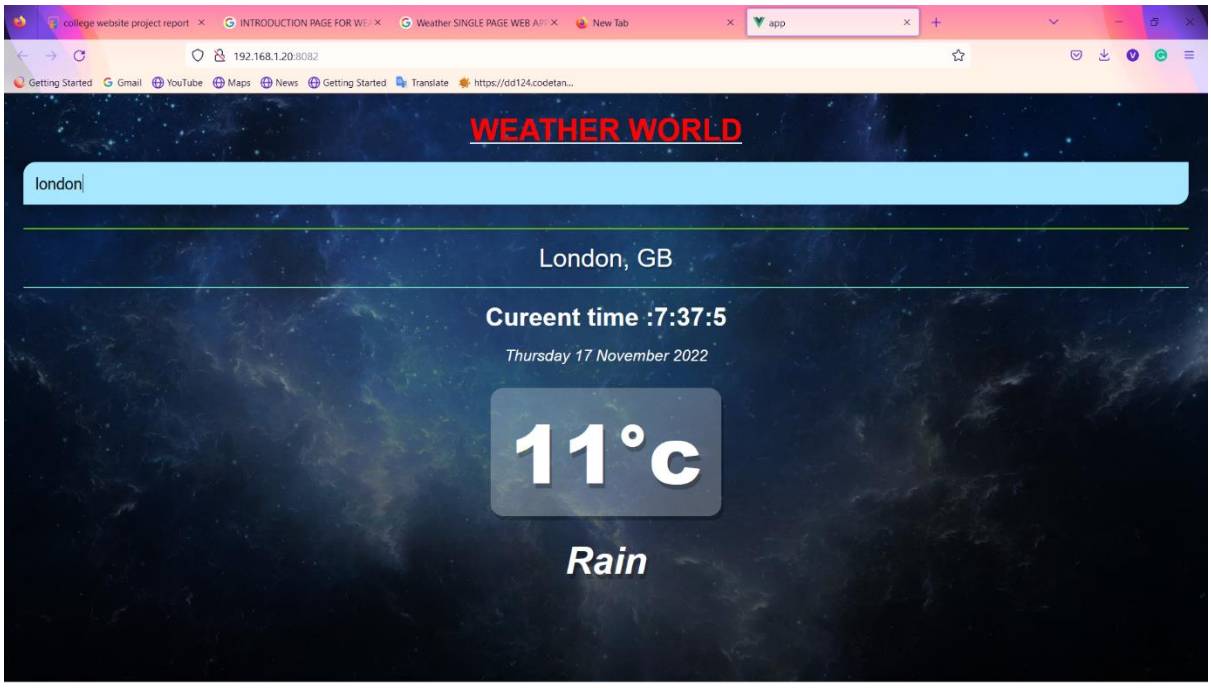
```
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).mount('#app')
```

SCREEN SHOTS OF MY CODE OUTPUT WEBSITE

:: First page when run



::After putting data temp<16(ex:London)-



::After putting the data temp>16(ex:Mumbai)-

WAYS TO REACH THE CODE SND INFORMATION ABOUT IT:-

Destination of the file-C: “\vuejs\vuewhether”

For creating the file command-:”Vue create weather”

About code:

I have created the element ‘’App’’.

For running the file through NPM command-: npm run serve

In this project I have used API of openweathermap.org and the API is-:

```
ff0148743305d28f98a6e79c11ea1c02"
```

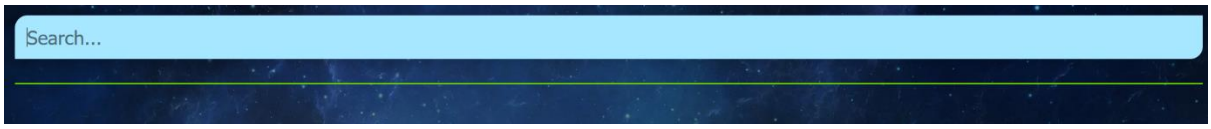
-:I have also calling the current time also from the server through getnow function and get hours and minutes too.

-:same I have done to date and day also from web.

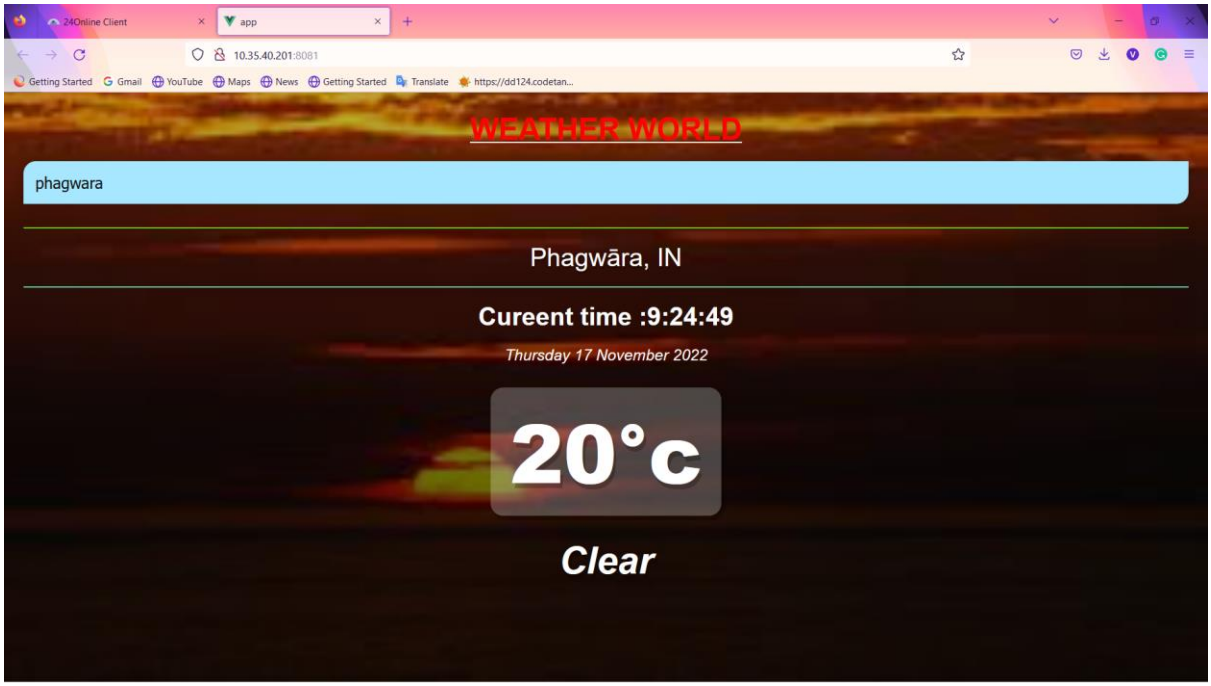
-:I have also made a TRANSITION in the search box on clicking it change the color .

Before clicking on it-:

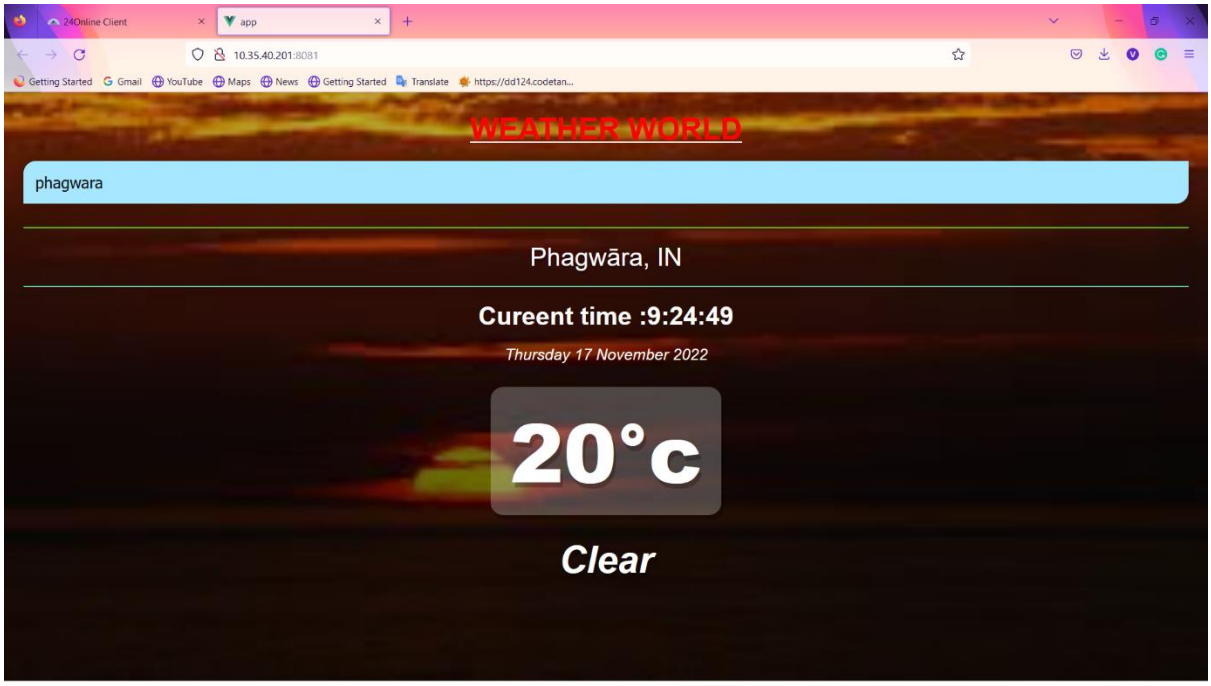
After clicking on it-:



-:I have changed the background with the change in temperature below 16 degree(temp<16):-



else temperature is above 16 (temp>16):



-:I have used different CSS properties also like -:

- margin

-padding

-box-sizing

-font-family

-background-image

-background -color

-height

-width

-appearance

-shadow

-transition

-text-align

-text-style

-font-weight

CONCLUSION-:

We have developed and set-up a low cost VUEJS based wireless automatic weather monitoring system with high accuracy.

The results are very good with a correlation factor greater than 0.85, while compared with the readings of Meteorological Centre and Snow and Avalanche Study Establishment Laboratory.

System is able to generate short term alerts and web hosting is successfully tested.

REFRENCES

-: You tube link: 1. https://youtu.be/JLc-hWsPTUY

2. https://youtu.be/5Uxe_MNd6go

-: GITHUB link :

1.https://github.com/TylerPottsDev/weather-vue

2.https://github.com/somteacodes/weatherappvue

-: Javatpoint:

https://www.javatpoint.com/vue-js

-:w3schools: https://www.w3schools.com/whatis/whatis_vue.asp

-:AND Vue js website: https://vuejs.org/