```css
h1{
    text-align: center;
    font-family: monospace;
    font-size: 8em;
    color: rgb(4, 63, 139);
}

page{
    background-color: rgb(152, 220, 236);
}

p{
    text-align: center;
    font-family: monospace;
    font-size: 2em;
    color: rgb(4, 63, 139);
}

input{
    text-align: center;
    font-family: monospace;
    font-size: 2em;
    color: rgb(0, 0, 0);
    margin: 0 auto;
    display: block;


}

button{
    text-align: center;
    font-family: monospace;
    font-size: 2em;
    color: rgb(4, 63, 139);
    margin: 0 auto;
    display: block;
    background-color: rgb(152, 220, 236);
}

img{
    display: block;
    margin: 0 auto;
    height: 300px;
    width: 400px;
}

table{
    margin: 0 auto;
    font-family: monospace;
    text-align: center;
    border: 5px solid black;
    border-spacing: 50px;
    font-size: 2em;
    color: rgb(4, 63, 139);

}
```

```
<script type="module">
    import { createApp } from "https://unpkg.com/vue@3/dist/vue.esm-browser.js"

    createApp({
        data() {
            return {
                cityName: null,
                umbrellaRequired: null,
                message: null,
                fact: null,
                healthMessage: null,
                avgTemp: null,
                avgRain: null,
                avgWind: null,
                pollutionWarning: null,
                temps: [],
                rains: [],
                winds: [],
                data: [],
            }
        },
        methods: {
            httpsGet(clicked) {
                if (clicked) {
                    fetch(`http://api.openweathermap.org/data/2.5/forecast?q=${this.cityName}&appid=b731b5a1b4a7d0c043bf6025ee0803bb&units=metric`)
                    .then((response) => response.json())
                    .then((cityData) => {
                        console.log(cityData);

                        //Show error if city is not inputted
                        if (cityData.cod == "404") {
                            this.message = "Please enter a valid city name";
                            this.fact = null;
                            this.healthMessage = null;
                            this.avgTemp = null;
                            this.avgRain = null;
                            this.avgWind = null;
                            this.pollutionWarning = null;
                            this.temps = [];
                            this.rains = [];
                            this.winds = [];
                            this.data = [];
                        } else {
                            this.message = null;
                            this.fact = cityData.list[0].weather[0].description;
                            this.healthMessage = null;
                            this.avgTemp = null;
                            this.avgRain = null;
                            this.avgWind = null;
                            this.pollutionWarning = null;
                            this.temps = [];
                            this.rains = [];
                            this.winds = [];
                            this.data = [];
                        }

                        //Packing: if there is rain anytime over the next 4-days indicate that the user should bring an umbrella.
                        this.umbrellaRequired = false;
                        for (let i = 0; i < cityData.list.length; i++) {
                            if (cityData.list[i].weather[0].main == "Rain") {
                                this.umbrellaRequired = true;
                            }
                        }
                        if (this.umbrellaRequired) {
                            this.message = "You should bring an umbrella.";
                        } else {
                            this.message = "You don't need an umbrella.";
                        }

                        //Give a summary table for the next 4 days showing: Temperature, Wind Speed and Rainfall level.
                        for (let i = 0; i < cityData.list.length; i++) {
                            this.temps.push(cityData.list[i].main.temp);
                            this.winds.push(cityData.list[i].wind.speed);
                            //push rain level to array
                            if (cityData.list[i].rain) {
                                this.rains.push(cityData.list[i].rain["3h"]);
                            } else {
                                this.rains.push(0);
                            }
                        }

                        //Indicate the type of weather that the user should pack for Cold (temperatures below12C), Mild (temperatures from 12C to 24C inclusive) or Hot (temperatures ab
                        this.avgTemp = 0;
                        for (let i = 0; i < cityData.list.length; i++) {
                            this.avgTemp += cityData.list[i].main.temp;
                        }
                        this.avgTemp = this.avgTemp / cityData.list.length;
                        if (this.avgTemp < 12) {
                            this.message += "It's cold. You should wear a coat.";
                        } else if (this.avgTemp > 24) {
                            this.message += " It's hot. You should not wear a coat.";
                        } else {
                            this.message += " It's mild. You either should or should not wear a coat.";
                        }


                        //Indicate if the city has a pollution level that is above 100 (i.e. the air quality is poor).
                        this.pollutionWarning = false;
                        fetch(`http://api.openweathermap.org/data/2.5/air_pollution/forecast?lat=${cityData.city.coord.lat}&lon=${cityData.city.coord.lon}&appid=b731b5a1b4a7d0c043bf602
                        .then((response) => response.json())
                        .then((pollutionData) => {
                            console.log(pollutionData);
                            for (let i = 0; i < pollutionData.list.length; i++) {
                                if (pollutionData.list[i].components.pm2_5 > 10) {
                                    this.pollutionWarning = true;
                                }
                            }
                            if (this.pollutionWarning) {
                                this.healthMessage = " The air quality is over 100. You should wear a mask.";
                            } else {
                                this.healthMessage = " The air quality is under 100. You don't need to wear a mask.";
                            }
                        })

                        //Use fetch API to get a historical event that happened in the city on the current date.
                        fetch(`https://api.api-ninjas.com/v1/historicalevents?text=${this.cityName}&date=${cityData.list[0].dt_txt}&limit=1`, {
                            headers: {
                                "X-Api-Key": "4w/ftYwAEPICdzNz4RoXyw==ALQm9QbIL0pVDzaK"
                            }
                        })
```

```
                        .then((response) => response.json())
                        .then((eventData) => {
                            console.log(eventData);
                            //return the date of the event
                            this.fact = "On this day in " + eventData[0].year + ", " + eventData[0].event;
                        })

                        //Use API to get an image of the selected city
                        fetch(`https://api.teleport.org/api/urban_areas/slug:${cityData.city.name.toLowerCase()}/images/`)
                        .then((response) => response.json())
                        .then((imageData) => {
                            console.log(imageData);
                            //return the image of the city
                            document.getElementById("cityImage").src = imageData.photos[0].image.web;
                        })



                    })

                }
            }
        }
    }).mount("#app")
</script>

<div id="app">

    <h1>Weather App</h1>
    <p>Enter a city name:</p>

    <!-- Input box for city name -->
    <input v-model="cityName" type="text" style="text-align:center">
    <button v-on:click="httpsGet(true)">Get Weather</button>
    <link rel="stylesheet" href="index.css">

    <!--Display the city name, country and current temperature!-->
    <p>{{message}}</p>
    <p>{{healthMessage}}</p>
    <p>{{fact}}</p>
    <img id="cityImage" src="https://i.pinimg.com/originals/77/0b/80/770b805d5c99c7931366c2e84e88f251.png" alt=" ">

    <!--Display a table of the next 4 days showing: Temperature, Wind Speed and Rainfall level.!-->
    <table>
        <tr>
            <th>     </th>
            <th>Day 1</th>
            <th>Day 2</th>
            <th>Day 3</th>
            <th>Day 4</th>
        </tr>
        <tr>
            <td>Average Temperature</td>
            <td>{{temps[0]}}°C</td>
            <td>{{temps[8]}}°C</td>
            <td>{{temps[16]}}°C</td>
            <td>{{temps[24]}}°C</td>
        </tr>
        <tr>
            <td>Wind Speed</td>
            <td>{{winds[0]}}m/s</td>
            <td>{{winds[8]}}m/s</td>
            <td>{{winds[16]}}m/s</td>
            <td>{{winds[24]}}m/s</td>
        </tr>
        <tr>
            <td>Rainfall</td>
            <td>{{rains[0]}}mm</td>
            <td>{{rains[8]}}mm</td>
            <td>{{rains[16]}}mm</td>
            <td>{{rains[24]}}mm</td>
        </tr>
    </table>
</div>
```