

INTRODUCTION

OVERVIEW :

Weather App is an one stop solution for staying up-to-date with real-time weather forecasts.

This project is an existing endeavor in "FRONT-END DEVELOPMENT" aimed to providing users with a sleek and intuitive weather application. Our mission is to deliver an engaging user experience by presenting weather data in a visually appealing and informative

PURPOSE :

Weather plays a significant role in our daily lives, influencing our purpose activities, clothing choices & overall well-being. People constantly seek accurate weather information to plan their schedules accordingly, while many weather applications exist. Weather App stands out by prioritizing user experience and simplicity.

The purpose of a weather App project is to create a software application that provide users with real-time weather information and forecasts for a specific location (or) for a multiple location.

PROPOSED SOLUTION :

WEATHER ALERTS :

The app may include a feature to send weather alerts and notifications to users for severe weather conditions

MAPS AND RADAR :

Including weather maps and radar data can help users visualize weather patterns and track storms

INTEGRATION WITH API'S :

The app may utilize third-party weather API's to access accurate and up-to-date weather data

CROSS - PLATFORM COMPATABILITY :

To reacher a boder audience , the app should be compatible with different operating systems such as android , ios , web browsers

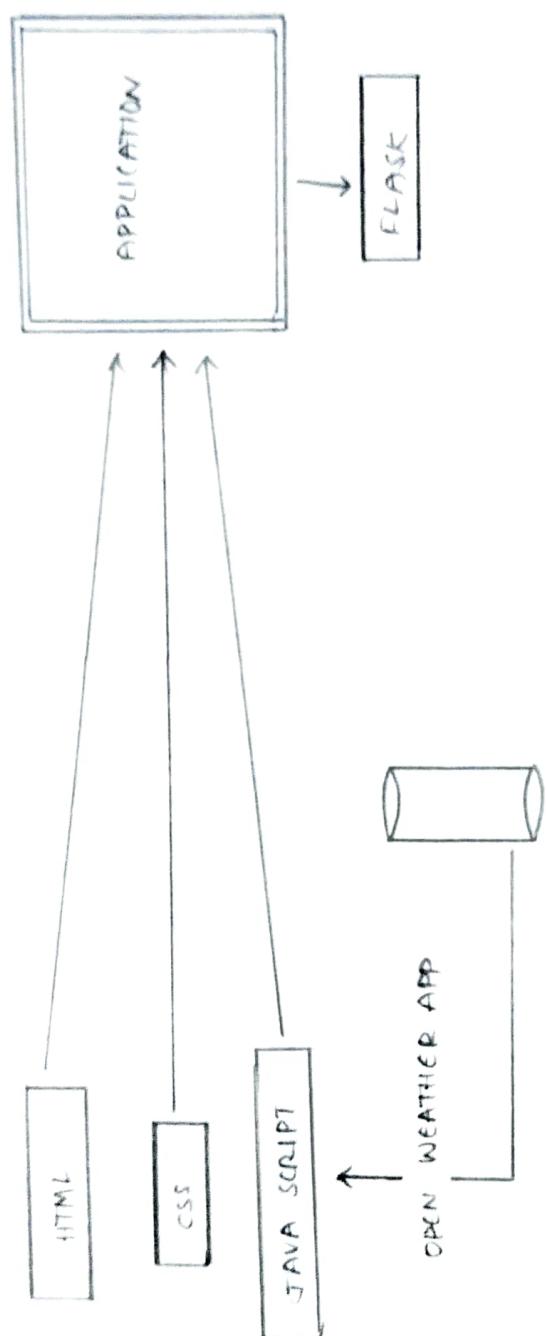
OFFLINE ACCESS :

Although real-time data is essential, the app might consider providing basic weather information even when the device is offline

Overall, the primary purpose of weather app project is to offer users a convenient and reliable tool to access weather information.

THEORETICAL ANALYSIS

BLOCK DIAGRAM :



LITERATURE SURVEY

EXISTING PROBLEM :

REAL - TIME WEATHER DATA :

The app should be able to fetch and display current weather conditions, including temperature, humidity, wind speed & visibility.

WEATHER FORECASTS :

Providing accurate weather forecasts for the next few days is crucial, as it helps users plan ahead for events, travel or outdoor activities.

LOCATION BASED SERVICES :

The App should be able to determine the user's location or allow them to input a specific location for weather information.

USER - FRIENDLY INTERFACE :

The App should have an intuitive and visually appealing interface, making it easy for users to understand and navigate.

CUSTOMIZATION :

Users may want to customize the app to display weather units in their preferred format.

HARDWARE / SOFTWARE DESIGNING :

Hardware and software requirements of the project

ACCESSING A DATA BASE :

- * the system should allow administrator to add historical weather data
- * The system should be able to recognize patterns in (s) temperature, humidity and wind with use of historical data

SOFTWARE CONSTRAINTS :

- * The development of the system will be constrained by the availability of required software such as web servers, dataset

HARDWARE REQUIREMENTS :

- * The system requires a database in order to store persistent data.

```
<!-->

<!DOCTYPE html>

<html lang="en">

  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Weather App</title>

    <!-- Stylesheet -->
    <link rel="stylesheet" href="style.css" />
  </head>

  <body>

    <div class="wrapper">
      <div class="container">
        <div class="search-container">
          <input
            type="text"
            placeholder="please enter a location"
            id="city"
            value="visakhapatnam"
          />
          <button id="search-btn">search</button>
        </div>
        <div id="result"></div>
        <div id="location-details">
          <button id="get-location"></button>
```

```
</div>

</div>

</div>

<!-- Script -->

<script src="script1.js"></script>

<script src="script.js"></script>

<script>

function autocomplete(inp, arr) {

    /*the autocomplete function takes two arguments,
    the text field element and an array of possible autocompleted values:*/
    var currentFocus;

    /*execute a function when someone writes in the text field:/

    inp.addEventListener("input", function(e) {

        var a, b, i, val = this.value;

        /*close any already open lists of autocompleted values/
        closeAllLists();

        if (!val) { return false; }

        currentFocus = -1;

        /*create a DIV element that will contain the items (values):/

        a = document.createElement("DIV");

        a.setAttribute("id", this.id + "autocomplete-list");

        a.setAttribute("class", "autocomplete-items");

        /*append the DIV element as a child of the autocomplete container:/

        this.parentNode.appendChild(a);

        /*for each item in the array.../
```

```
for (i = 0; i < arr.length; i++) {  
  
    /check if the item starts with the same letters as the text field value:/  
  
    if (arr[i].substr(0, val.length).toUpperCase() == val.toUpperCase()) {  
  
        /create a DIV element for each matching element:/  
  
        b = document.createElement("DIV");  
  
        /make the matching letters bold:/  
  
        b.innerHTML = "<strong>" + arr[i].substr(0, val.length) + "</strong>";  
  
        b.innerHTML += arr[i].substr(val.length);  
  
        /insert a input field that will hold the current array item's value:/  
  
        b.innerHTML += "<input type='hidden' value='" + arr[i] + "'>";  
  
        /execute a function when someone clicks on the item value (DIV element):/  
  
        b.addEventListener("click", function(e) {  
  
            /insert the value for the autocomplete text field:/  
  
            inp.value = this.getElementsByTagName("input")[0].value;  
  
            /*close the list of autocompleted values,  
             (or any other open lists of autocompleted values:*/  
  
            closeAllLists();  
  
        });  
  
        a.appendChild(b);  
  
    }  
  
};  
  
/execute a function presses a key on the keyboard:/  
  
inp.addEventListener("keydown", function(e) {  
  
    var x = document.getElementById(this.id + "autocomplete-list");
```

```
if (x) x = x.getElementsByTagName("div");
if (e.keyCode == 40) {
    /*If the arrow DOWN key is pressed,
    increase the currentFocus variable:*/
    currentFocus++;
    /and and make the current item more visible:/
    addActive(x);
} else if (e.keyCode == 38) { //up
    /*If the arrow UP key is pressed,
    decrease the currentFocus variable:*/
    currentFocus--;
    /and and make the current item more visible:/
    addActive(x);
} else if (e.keyCode == 13) {
    //If the ENTER key is pressed, prevent the form from being submitted,
    e.preventDefault();
    if (currentFocus > -1) {
        /and simulate a click on the "active" item:/
        if (x) x[currentFocus].click();
    }
}
});

function addActive(x) {
```

```
/a function to classify an item as "active":/
if (!x) return false;

/start by removing the "active" class on all items:/
removeActive(x);

if (currentFocus >= x.length) currentFocus = 0;

if (currentFocus < 0) currentFocus = (x.length - 1);

/add class "autocomplete-active":/
x[currentFocus].classList.add("autocomplete-active");

}

function removeActive(x) {
/a function to remove the "active" class from all autocomplete items:/
for (var i = 0; i < x.length; i++) {
x[i].classList.remove("autocomplete-active");
}

}

function closeAllLists(elmnt) {
/*close all autocomplete lists in the document,
except the one passed as an argument*/
var x = document.getElementsByClassName("autocomplete-items");
for (var i = 0; i < x.length; i++) {
if (elmnt != x[i] && elmnt != inp) {
x[i].parentNode.removeChild(x[i]);
}
}
}
```

/execute a function when someone clicks in the document:/

```
document.addEventListener("click", function (e) {  
    closeAllLists(e.target);  
});  
}  
  
}
```

/An array containing all the country names in the world:/

```
var countries =  
["adoni","Amaravati","Anantapur","Chandragiri","Chittoor","Dowlaiswaram","Eluru","Guntur","Kadapa"  
, "Kakinada","Kurnool","Machilipatnam","Nagarjunakonda","Rajahmundry","Srikakulam","Tirupati","Vija  
yawada","Visakhapatnam","Vizianagaram","Yemmiganur","Arunachal  
Pradesh","Itanagar","Assam","Dhuburi","Dibrugarh","Dispur","Guwahati","Jorhat","Nagaon","Sivasagar"  
, "silchar","Tezpur","Tinsukia","Bihar","Ara","Barauni","Begusarai","Bettiah","Bhagalpur","Bihar  
Sharif","Bodh Gaya","Buxar","Chapra","Darbhanga","Dehri","Dinapur  
Nizamat","Gaya","Hajipur","Jamalpur","Katihar","Madhubani","Motihari","Munger","Muzaffarpur","Pat  
na","Purnia","Pusa","Saharsa","Samastipur","Sasaram","Sita","marhi","Siwan","Chandigarh (union  
territory)","Chandigarh","Chhattisgarh","Ambikapur","Bhilai","Bilaspur","Dhamtari","Durg","Jagdalpur",  
"Raipur","Rajnandgaon","Dadra","Daman","Diu","Silvassa","New  
Delhi","Goa","Madgaon","Panaji","Gujarat","Ahmadabad","Amreli","Bharuch","Bhavnagar","Bhuj","Dwa  
rka","Gandhinagar","Godhra","Jamnagar","Junagadh","Kandla","Khambhat","Kheda","Mahesana","Mor  
bi","Nadiad","Navsari","Okha","Palanpur","Patan","Porbandar","Rajkot","Surat","Surendranagar","Vals  
d","Veraval","Haryana","Ambala","Bhiwani","Chandigarh","Faridabad","Firozpur  
Jhirka","Gurugram","Hansi","Hisar","Jind","Kaithal","Karnal","Kurukshestra","Panipat","Pehowa","Rewari  
","Rohtak","Sirsa","Sonipat","Himachal  
Pradesh","Bilaspur","Chamba","Dalhousie","Dharmshala","Hamirpur","Kangra","Kullu","Mandi","Nahan  
, "Shimla","Una","Anantnag","Baramula","Doda","Gulmarg","Jammu","Kathua","Punch","Rajouri","Srin  
agar","Udhampur","Jharkhand","Bokaro","Chaibasa","Deoghar","Dhanbad","Dumka","Giridih","Hazarib  
ag","Jamshedpur","Jharia","Rajmahal","Ranchi","Saraikela","Karnataka","Badami","Ballari","Bengaluru",  
"Belagavi","Bhadrapur","Bidar","Chikkamagaluru","Chitradurga","Davangere","Halebid","Hassan","Hub  
balli-  
Dharwad","Kalaburagi","Kolar","Madikeri","Mandy","Mangaluru","Mysuru","Raichur","Shivamogga","S  
hravanabelagola","Shrirangapattana","Tumakuru","Vijayapura","Kerala","Alappuzha","Vatakara","Idukki  
,"Kannur","Kochi","Kollam","Kottayam","Kozhikode","Mattancheri","Palakkad","Thalassery","Thiruvana  
nthapuram","Thrissur","Ladakh","Kargil","Leh","Madhya  
Pradesh","Balaghat","Barwani","Betul","Bharhut","Bhind","Bhojpur","Bhopal","Burhanpur","Chhatarpur  
, "Chhindwara","Damoh","Datia","Dewas","Dhar","Dr. Ambedkar  
,"Nagar","Guna","Gwalior","Hoshangabad","Indore","Itarsi","Jabalpur","Jhabua","Khajuraho","Khandwa",  
"Nagpur","Raigarh","Raigarh","Raigarh"]
```

"Khargone", "Maheshwar", "Mandla", "Mandsaur", "Morena", "Murwara", "Narsimhapur", "Narsinghgarh", "Narwar", "Neemuch", "Nowgong", "Orchha", "Panna", "Raisen", "Rajgarh", "Ratlam", "Rewa", "Sagar", "Saranpur", "Satna", "Sehore", "Seoni", "Shahdol", "Shajapur", "Sheopur", "Shivpuri", "Ujjain", "Vidisha", "Maharashtra", "Ahmadnagar", "Akola", "Amravati", "Aurangabad", "Bhandara", "Bhusawal", "Bid", "Buldhana", "Chandrapur", "Daulatabad", "Dhule", "Jalgaon", "Kalyan", "Karli", "Kolhapur", "Mahabaleshwar", "Malegaon", "Madrar", "Mumbai", "Nagpur", "Nanded", "Nashik", "Osmanabad", "Pandharpur", "Parbhani", "Pune", "Ratnatheran", "Sangli", "Satara", "Sevagram", "Solapur", "Thane", "Ulhasnagar", "Vasai-giri", "Wardha", "Yavatmal", "Manipur", "Imphal", "Meghalaya", "Cherrapunji", "Shillong", "Mizoram", "Aizavirar", "Lunglei", "Nagaland", "Kohima", "Mon", "Phek", "Wokha", "Zunheboto", "Odisha", "Balangir", "Baleshwari", "Baripada", "Bhubaneshwar", "Brahmapur", "Cuttack", "Dhenkanal", "Kendujhar", "Konark", "Koraput", "Par", "Phulabani", "Puri", "Sambalpur", "Udayagiri", "Puducherry", "Karaikal", "Mahe", "Puducherry", "Yanaradip", "Punjab", "Amritsar", "Batala", "Chandigarh", "Faridkot", "Firozpur", "Gurdaspur", "Hoshiarpur", "Jalandam", "Ludhiana", "Nabha", "Patiala", "Rupnagar", "Sangrur", "Rajasthan", "Abu", "Ajmer", "Alwar", "Kapurthala", "Kota", "Merta", "Nagaur", "Nathdwara", "Pali", "Phalodi", "Pushkar", "Sawaiodhpur", "Kishangarh", "Tonk", "Udaipur", "Sikkim", "Gangtok", "Gyalshing", "Lachung", "Madhopur", "Shahpura", "Sikar", "Sirohi", "Uttar", "Tamil Nadu", "Arcot", "Chengalpattu", "Chennai", "Chidambaram", "Coimbatore", "Cuddalore", "Dharmapuri", "Din", "Erode", "Kanchipuram", "Kanniyakumari", "Kodaikanal", "Kumbakonam", "Madurai", "Mamallapuradigul", "Nagercoil", "Palayamkottai", "Pudukkottai", "Rajapalayam", "Ramanathapuram", "Sam", "Nagappattinam", "Thanjavur", "Tiruchirappalli", "Tirunelveli", "Tiruppur", "Thoothukudi", "Udhagamandalam", "Vellorem", "Telangana", "Hyderabad", "Karimnagar", "Khammam", "Mahbubnagar", "Nizamabad", "Sangareddi", "We", "Tripura", "Agartala", "Uttar Pradesh", "Agra", "Aligarh", "Amroha", "Ayodhya", "Azamgarh", "Bahrach", "Ballia", "Banda", "Barabanki", "Bareilly", "Basti", "Bijnor", "Bithur", "Budaun", "Bulandshahr", "Deoria", "Etah", "Etawah", "Faizabad", "Farrukhabad-cum-Fatehgarh", "Fatehpur", "Fatehpur Sikri", "Ghaziabad", "Ghazipur", "Gonda", "Gorakhpur", "Hamirpur", "Hardoi", "Hathras", "Jalaun", "Jaunpur", "Jhansi", "Kannauj", "Kanpur", "Lakhimpur", "Lalitpur", "Lucknow", "Mainpuri", "MathuraMeerut", "Mirzapur-VindhyaChal", "Moradabad", "Muzaffarnagar", "Partapgarh", "Pilibhit", "Prayagraj", "Rae Bareli", "Rampur", "Saharanpur", "Sambhal", "Shahjahanpur", "Sitapur", "Sultanpur", "Tehri", "Varanasi", "Utarakhand", "Almora", "Dehra Dun", "Haridwar", "Mussoorie", "Nainital", "Pithoragarh", "West Bengal", "Alipore", "Alipur Duar", "Asansol", "Baharampur", "Bally", "Balurghat", "Bankura", "Baranagar", "Barasat", "Barrackpore", "Basilhat", "Bhatpara", "Bishnupur", "Budge Budge", "Burdwan", "Chandernagore", "Darjeeling", "Diamond Harbour", "Dum Dum", "Durgapur", "Halisahar", "Haora", "Hugli", "Ingraj Bazar", "Jalpaiguri", "Kalimpong", "Kamarhati", "Kanchrapara", "Kharagpur", "Cooch Behar", "Kolkata", "Krishnanagar", "Malda", "Midnapore", "Murshidabad", "Nabadwip", "Palashi", "Panighati", "Purulia", "Raiganj", "Santipur", "Shantiniketan", "Shrirampur", "Siliguri", "Siuri", "Titagarh", "Tamluk", "Afghanistan", "Albania", "Algeria", "Andorra", "Angola", "Anguilla", "Antigua & Barbuda", "Argentina", "Armenia", "Aruba", "Australia", "Austria", "Azerbaijan", "Bahamas", "Bahrain", "Bangladesh", "Barbados", "Belarus", "Belgium", "Belize", "Benin", "Bermuda", "Bhutan", "Bolivia", "Bosnia &

Herzegovina", "Botswana", "Brazil", "British Virgin Islands", "Brunei", "Bulgaria", "Burkina Faso", "Burundi", "Cambodia", "Cameroon", "Canada", "Cape Verde", "Cayman Islands", "Central African Republic", "Chad", "Chile", "China", "Colombia", "Congo", "Cook Islands", "Costa Rica", "Cote D Ivoire", "Croatia", "Cuba", "Curacao", "Cyprus", "Czech Republic", "Denmark", "Djibouti", "Dominica", "Dominican Republic", "Ecuador", "Egypt", "El Salvador", "Equatorial Guinea", "Eritrea", "Estonia", "Ethiopia", "Falkland Islands", "Faroe Islands", "Fiji", "Finland", "France", "French Polynesia", "French West Indies", "Gabon", "Gambia", "Georgia", "Germany", "Ghana", "Gibraltar", "Greece", "Greenland", "Grenada", "Guam", "Guatemala", "Guernsey", "Guinea", "Guinea Bissau", "Guyana", "Haiti", "Honduras", "Hong Kong", "Hungary", "Iceland", "India", "Indonesia", "Iran", "Iraq", "Ireland", "Isle of Man", "Israel", "Italy", "Jamaica", "Japan", "Jersey", "Jordan", "Kazakhstan", "Kenya", "Kiribati", "Kosovo", "Kuwait", "Kyrgyzstan", "Laos", "Latvia", "Lebanon", "Lesotho", "Liberia", "Libya", "Liechtenstein", "Lithuania", "Luxembourg", "Macau", "Macedonia", "Madagascar", "Malawi", "Malaysia", "Maldives", "Mali", "Malta", "Marshall Islands", "Mauritania", "Mauritius", "Mexico", "Micronesia", "Moldova", "Monaco", "Mongolia", "Montenegro", "Montserrat", "Morocco", "Mozambique", "Myanmar", "Namibia", "Nauro", "Nepal", "Netherlands", "Netherlands Antilles", "New Caledonia", "New Zealand", "Nicaragua", "Niger", "Nigeria", "North Korea", "Norway", "Oman", "Pakistan", "Palau", "Palestine", "Panama", "Papua New Guinea", "Paraguay", "Peru", "Philippines", "Poland", "Portugal", "Puerto Rico", "Qatar", "Reunion", "Romania", "Russia", "Rwanda", "Saint Pierre & Miquelon", "Samoa", "San Marino", "Sao Tome and Principe", "Saudi Arabia", "Senegal", "Serbia", "Seychelles", "Sierra Leone", "Singapore", "Slovakia", "Slovenia", "Solomon Islands", "Somalia", "South Africa", "South Korea", "South Sudan", "Spain", "Sri Lanka", "St Kitts & Nevis", "St Lucia", "St Vincent", "Sudan", "Suriname", "Swaziland", "Sweden", "Switzerland", "Syria", "Taiwan", "Tajikistan", "Tanzania", "Thailand", "Timor L'Este", "Togo", "Tonga", "Trinidad & Tobago", "Tunisia", "Turkey", "Turkmenistan", "Turks & Caicos", "Tuvalu", "Uganda", "Ukraine", "United Arab Emirates", "United Kingdom", "United States of America", "Uruguay", "Uzbekistan", "Vanuatu", "Vatican City", "Venezuela", "Vietnam", "Virgin Islands (US)", "Yemen", "delhi", "Zambia", "Zimbabwe"];

/initiate the autocomplete function on the "myInput" element, and pass along the countries array as possible autocomplete values:/

```
autocomplete(document.getElementById("city"), countries);

</script>

</body>

</html>
```

```
//  
  
* {  
    padding: 0;  
    margin: 0;  
    box-sizing: border-box;  
    font-family: 'Times New Roman', Times, serif;  
    color: black;  
}  
  
.search-container #city{  
    color: aliceblue;  
}  
  
}  
  
:root {  
    --white: #ffffff;  
    --off-white: #e5e5e5;  
    --transp-white-1: rgba(255, 255, 255, 0.25);  
    --transp-white-2: rgba(255, 255, 255, 0.1);  
    --green: #0c0836;  
    --blue: #0b0b0c;
```

```
--shadow: rgba(2, 28, 53, 0.2);  
}  
  
body {  
    height: 50vh;  
    background-image:  
        url(images/Inkedbackground.png);  
}  
  
.wrapper {  
    font-size: 16px;  
    width: 90vw;  
    max-width: 28em;  
    position: absolute;  
    transform: translate(-50%, -50%);  
    top: 50%;  
    left: 50%;  
}
```

```
.container {  
    width: 100%;  
    background: var(--green);  
    backdrop-filter: blur(10px);  
    padding: 3em 1.8em;  
    border: 2px solid var(--transp-white-2);  
    border-radius: 0.6em;  
    box-shadow: 0 1.8em 3.7em var(--shadow);  
    text-align: center;  
    margin-left: -20rem;  
}  
  
.search-container {  
    font-size: 1em;  
    display: grid;  
    grid-template-columns: 9fr 3fr;  
    gap: 1.25em;  
}
```

```
.search-container input,  
.search-container button {  
    outline: none;  
    font-size: 1em;  
    border: none;  
}  
  
.search-container input {  
    padding: 0.7em;  
    background-color: transparent;  
    border-bottom: 2px solid var(--transp-white-1);  
    color:white;  
    font-weight: 300;  
}  
  
.search-container input::placeholder {  
    color: var(--off-white);  
}  
  
.search-container input:focus {
```

```
border-color: var(--white);  
}  
.search-container button {  
color: var(--blue-2);  
background-color: var(--white);  
border-radius: 0.3em;  
}  
  
#result h2 {  
color: var(--white);  
text-transform: uppercase;  
letter-spacing: 0.18em;  
font-weight: 600;  
margin: 1.25em 0;  
}  
  
.weather,  
.desc {  
color: var(--off-white);
```

```
text-transform: uppercase;  
letter-spacing: 0.2em;  
font-size: 0.9em;  
font-weight: 500;  
line-height: 2em;  
}  
  
.weather {  
margin-top: -0.7em;  
}  
  
#result img {  
margin: 0.6em 0 0 0;  
width: 6.2em;  
filter: drop-shadow(0 1.8em 3.7em var(--shadow));  
}  

```

```
line-height: 0;  
font-weight: 400;  
color: var(--white);  
}  
  
.temp-container {  
display: flex;  
justify-content: center;  
}  
  
.temp-container div {  
padding: 0.3em 1em;  
}  
  
.temp-container div:first-child {  
border-right: 1px solid var(--transp-white-1);  
}  
  
.temp-container .title {  
font-weight: 600;  
color: var(--white);
```

```
}

.temp-container .temp {
    font-weight: 300;
    color: var(--off-white);
}

.msg {
    margin-top: 1.8em;
    color: var(--white);
    font-weight: 500;
    text-transform: uppercase;
    letter-spacing: 0.1em;
}

#location-details{
    height: 2px;
    width: px;
    color: antiquewhite;
}
```

```
#get-location img{  
    height: 30px;  
    width: 20px;  
    position: absolute;  
    bottom: 350px;  
    left: 50px;  
}  
  
@media screen and (max-width: 450px) {  
    .wrapper {  
        font-size: 14px;  
    }  
}
```

```
let result = document.getElementById("result");

let searchBtn = document.getElementById("search-btn");

let cityRef = document.getElementById("city");

const key = "1946852246b6fe5011a8a256564b9718";

//Function to fetch weather details from api and display them

let getWeather = () => {

  let cityValue = cityRef.value;

  //If input field is empty

  if (cityValue.length == 0) {

    result.innerHTML = `<h3 class="msg">Please enter a city
name</h3>`;

  }

  //If input field is NOT empty

  else {

    let url =
`https://api.openweathermap.org/data/2.5/weather?q=${cityValue}
&appid=${key}&units=metric`;

    //Clear the input field

    cityRef.value = "";
  }
}
```

```
fetch(url)

.then((resp) => resp.json())

//If city name is valid

.then((data) => {

    console.log(data);

    console.log(data.weather[0].icon);

    console.log(data.weather[0].main);

    console.log(data.weather[0].description);

    console.log(data.name);

    console.log(data.main.temp_min);

    console.log(data.main.temp_max);

    result.innerHTML = `

<h2>${data.name}</h2>

<h4 class="weather">${data.weather[0].main}</h4>

<h4 class="desc">${data.weather[0].description}</h4>



<h1>${data.main.temp} &#176;</h1>

<div class="temp-container">
```

```
<div>

    <h4 class="title">min</h4>
    <h4 class="temp">${data.main.temp_min}°</h4>
</div>

<div>
    <h4 class="title">max</h4>
    <h4 class="temp">${data.main.temp_max}°</h4>
</div>

</div>
`;

})

//If city name is NOT valid

.catch(() => {
    result.innerHTML = `<h3 class="msg">City not found</h3>`;
});

}

};

searchBtn.addEventListener("click", getWeather);

window.addEventListener("load", getWeather);

let locationButton = document.getElementById("get-location");
```

```
let locationDiv = document.getElementById("location-details");

locationButton.addEventListener("click", () => {
    //Geolocation API is used to get geographical position of a user and
    //is available inside the navigator object
    if (navigator.geolocation) {
        //returns position(latitude and longitude) or error
        navigator.geolocation.getCurrentPosition(showLocation,
        checkError);
    } else {
        //For old browser i.e IE
        locationDiv.innerText = "The browser does not support
        geolocation";
    }
});

//Error Checks
const checkError = (error) => {
    switch (error.code) {
        case error.PERMISSION_DENIED:
            locationDiv.innerText = "Please allow access to location";
    }
}
```

```
break;

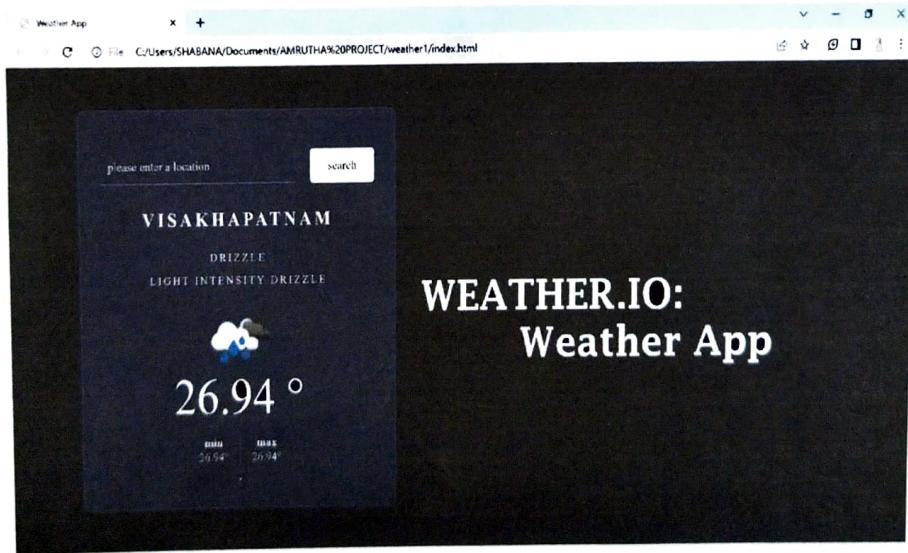
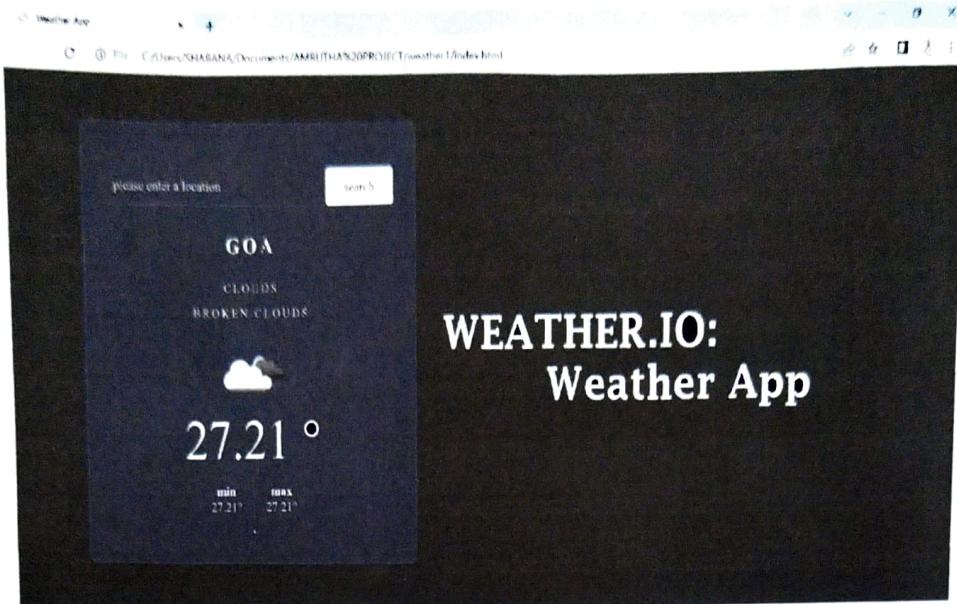
case error.POSITION_UNAVAILABLE:
    //usually fired for firefox
    locationDiv.innerText = "Location Information unavailable";
    break;

case error.TIMEOUT:
    locationDiv.innerText = "The request to get user location timed
out";
}

};

const showLocation = async (position) => {
    //We user the Nominatim API for getting actual addres from latitude
    and longitude
    let response = await fetch(
        `https://nominatim.openstreetmap.org/reverse?lat=${position.coord
s.latitude}&lon=${position.coords.longitude}&format=json`
    );
    //store response object
    let data = await response.json();
```

```
locationDiv.innerText = `${data.address.city},  
`${data.address.country}`;
```



ADVANTAGES AND DISADVANTAGES

ADVANTAGES :

SKILL ENHANCEMENT :

Developing a weather app as a front-end project allows front end developers to improve their skills in HTML, CSS and Javascript

REAL - WORLD APPLICATION :

A Weather app is a practical project that provides real-world value to users. It also allows developers to work on something relevant.

USER INTERFACE DESIGN :

Weather apps require an intuitive and usually appealing user interface. Building such an interface helps to sharpen their design

RELIANCE ON TECHNOLOGY :

Weather forecasting relies heavily on technology and if the technology fails or is unavailable, accurate predictions cannot be made

LIMITED TIME FRAME :

Forecasts are usually only accurate for a short time frame, making it difficult to plan ahead

DISADVANTAGES :

DATA LIMITATIONS :

Front-end developers rely on weather API's to fetch weather data. The amount of data and the available features are dependent.

LACK OF BACKEND EXPERIENCE :

Building a weather app purely as a front-end project may not provide opportunities to gain experience in server-side programming.

SECURITY CONCERN :

Handling API's and external data sources requires careful consideration of security to prevent data breaches to sensitive information.

CONFUSING TERMINOLOGY :

The terminology used in weather forecasting can be confusing, making it difficult for some people to understand predictions.

LIMITED REACH :

Weather forecasts are not available for many remote or sparsely populated areas, making it difficult for people in these areas to prepare for severe weather.

APPLICATIONS

REAL-TIME WEATHER INFORMATION :

Display current weather conditions, including temperature, humidity, wind speed and direction, along with a icon representing weather type

Eg: sunny, cloudy, Rainy

LOCATION - BASED FORECAST :

Allow users to enter their location or use their devices gps to get localized weather forecasts for the current day and upcoming days

MULTIPLE LOCATIONS :

Enable users to save and switch between multiple location, so they can check weather for places they frequently visited

WEATHER RADAR AND MAPS :

Implement weather radar and interactive maps to visualize weather patterns including rain, snow and cloud cover

WEATHER ALERTS AND WARNINGS :

Display severe weather alerts & warnings for the users locations or selected regions, ensuring users stay informed about potentially dangerous conditions.

HOURLY AND DAILY FORECASTS :

provide detailed weather forecasts for the next few hours and several days ahead , giving users a comprehensive view of what to expect

USER PREFERENCES :

let users customize the app by setting the temperature units

e.g: celsius (or) fahrenheit

HISTORICAL WEATHER DATA :

offer access to historical weather data, allowing user to explore past weather patterns

SOCIAL MEDIA INTEGRATION :

allow users to share weather updates on social media platforms

RESPONSIVE DESIGNS :

ensure the app is fully responsive and optimized for various devices

ACCESSIBILITY :

make the app accessible to users with disabilities by gathering to accessibility standards and guidelines

CONCLUSION

The weather apps are increasingly accurate and useful, and their benefits extend widely across the economy. While much has been accomplished in improving weather forecast, there remains much room for improvements.

simultaneously, they are developing new technologies and observational networks that can enhance forecast skill.

FUTURE SCOPE

The demand for weather and climate forecast information in support of critical decision making has grown rapidly during the last decade, and will grow even faster in the coming years. Great advances have been made in the utilization of predictions in many areas of human activities.

The future of weather applications is promising, with the increasing demand for real time and accurate weather information.