

WEATHER TRACKING APPLICATION

A PROJECT REPORT

Submitted by

SHANTHOSH S (220701263)

in partial fulfillment for the course

CS19542 – INTERNET PROGRAMMING

for the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND DESIGN

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR

THANDALAM

CHENNAI – 602 105

NOVEMBER 2024

RAJALAKSHMI ENGINEERING COLLEGE

CHENNAI - 602105

BONAFIDE CERTIFICATE

Certified that this project report “**Weather Tracking Application**” is the bonafide work of “**SHANTHOSH S(220701263)**” who carried out the project work for the subject CS19542 – Internet Programming under my supervision.

SIGNATURE

Mr. DeepakKumar M

SUPERVISOR

Associate Professor

Department of

Computer Science and Engineering

Rajalakshmi Engineering College

Rajalakshmi Nagar

Thandalam

Chennai – 602105

Submitted to Project and Viva Voce Examination for the subject

CS19542 – Internet Programming held on 25 – 11 - 2024

Abstract:

In the current global scenario, climate awareness and preparedness have become critical aspects of daily life. With unpredictable weather patterns and increasing environmental challenges, access to accurate and timely weather information is essential for personal safety, efficient planning, and fostering a sustainable lifestyle. There is a growing demand for innovative solutions that provide real-time, reliable, and accessible weather updates, enabling individuals to make informed decisions. The Weather Tracking Application addresses these challenges by offering a comprehensive, user-friendly platform for real-time weather monitoring. Equipped with advanced features like dynamic search, current location-based tracking, and detailed metrics including temperature, humidity, air quality, and wind speed, the application ensures precise and actionable insights. Users can access hourly updates, 5-day forecasts, and environmental highlights such as sunrise and sunset timings. Designed with a responsive interface and powered by efficient APIs, the application provides seamless access across devices. By bridging the gap between complex meteorological data and user needs, this system stands as a vital tool in enhancing climate preparedness and promoting informed decision-making.

CHAPTER 1

Introduction:

Weather plays a crucial role in shaping human activities, from daily routines to long-term planning. As climate change continues to intensify and disrupt traditional weather patterns, the need for accurate, real-time weather tracking has never been greater. Reliable weather information is essential for ensuring safety, mitigating risks, and making informed decisions, especially in sectors like agriculture, transportation, disaster management, and tourism. Despite the availability of numerous weather services, many existing systems are either outdated or provide limited data that fails to meet the diverse needs of modern users. This gap highlights the importance of developing advanced tools that integrate real-time updates, detailed forecasts, and user-centric design. The Weather Tracking Application has been conceptualized and developed to address these challenges. Designed to provide accurate, real-time weather information for locations worldwide, this application ensures accessibility, ease of use, and detailed insights. With features such as current location-based tracking, dynamic search, and comprehensive data visualization, the platform offers a modern solution to weather monitoring, empowering users to adapt to changing environmental conditions effectively.

Objective:

The primary objective of the Weather Tracking Application is to provide users with accurate, real-time weather information and forecasts in a user-friendly and accessible manner. The application aims to:

1. **Enhance Climate Awareness:** Equip users with comprehensive weather data, enabling them to stay informed about changing weather conditions and plan their activities accordingly.
2. **Provide Real-Time Insights:** Offer up-to-the-minute updates on temperature, humidity, wind speed, air quality, visibility, and other key metrics to ensure reliable information.
3. **Simplify Weather Tracking:** Deliver an intuitive interface with features like location-based tracking, dynamic search, and seamless data visualization for an effortless user experience.
4. **Support Decision-Making:** Enable users to make informed decisions in their personal and professional lives by providing detailed 5-day forecasts and hourly updates.
5. **Promote Accessibility:** Ensure that the application is responsive and accessible across various devices, allowing users to monitor weather anytime, anywhere.

Through these objectives, the application aims to bridge the gap between complex meteorological data and user needs, fostering a culture of preparedness and awareness in an increasingly unpredictable climate.

Existing System:

Numerous weather tracking systems and applications are currently available, offering varying levels of service. Popular platforms like AccuWeather, Weather.com, and regional meteorological services provide users with access to basic weather forecasts and current conditions. These systems often include additional features such as radar maps, severe weather alerts, and climate news, catering to a wide audience ranging from casual users to professionals. Despite their widespread use, existing solutions face several limitations. Many

applications rely on outdated interfaces or cluttered designs, making it difficult for users to navigate and access relevant information quickly. Additionally, the accuracy of weather data can vary depending on the sources used and the frequency of updates. Some services also lack advanced features, such as detailed air quality indices, precise location-based tracking, or interactive data visualizations, which are increasingly demanded by users. Furthermore, the availability of features across devices remains inconsistent, with some platforms not optimized for mobile or tablet use. These shortcomings highlight the need for more user-centric and technologically advanced solutions that prioritize real-time accuracy, accessibility, and ease of use, paving the way for the development of innovative tools like the Weather Tracking Application.

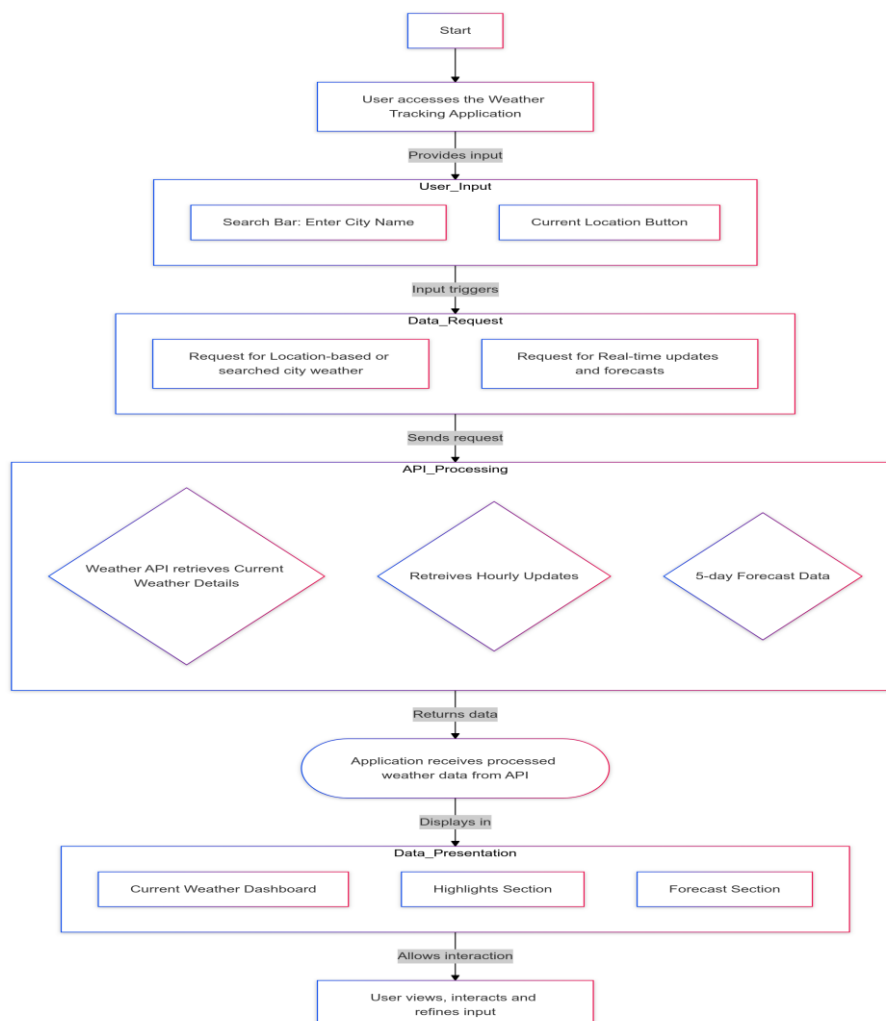
Proposed Solution:

The Weather Tracking Application offers a modern and efficient solution to address the limitations of existing weather tracking systems. It is designed to provide accurate, real-time weather data with an intuitive and accessible interface, catering to the diverse needs of users. The application leverages advanced APIs to deliver precise weather metrics such as temperature, humidity, air quality, wind speed, and visibility for any location worldwide. With features like dynamic search and current location-based tracking, users can easily access weather information tailored to their needs. The application also includes detailed 5-day forecasts and hourly updates, enabling users to plan effectively for upcoming weather changes. To enhance user experience, the system incorporates responsive design principles, ensuring seamless accessibility across devices, including desktops, tablets, and smartphones. Interactive data visualizations and concise highlights, such as sunrise and sunset timings, further simplify the interpretation of weather data.

CHAPTER 2

System Flow:

The system flow of the Weather Tracking Application begins with the user accessing the platform and providing input either through a search bar or by enabling location-based tracking. The application then sends a data request to the weather API, which processes and retrieves real-time weather details, including current conditions, hourly updates, and 5-day forecasts. The received data is displayed on an intuitive user interface, organized into a current weather dashboard, highlights, and forecast sections. Users can interact with the information, refine their input, or refresh for updates, ensuring a seamless and informative weather tracking experience.



CHAPTER 3

Project Description:

The Weather Tracking Application is a web-based platform designed to provide users with accurate, real-time weather information for locations worldwide.

Developed with modern web technologies, the application ensures a seamless user experience through its intuitive interface and responsive design, catering to a wide range of devices, including desktops, tablets, and smartphones. The application allows users to search for weather data by city or leverage location-based tracking for precise updates. It provides a comprehensive view of current weather conditions, including temperature, humidity, air quality index, wind speed, visibility, and sunrise and sunset timings. Users also have access to hourly updates and a detailed 5-day forecast, enabling them to plan ahead effectively. To ensure accuracy and reliability, the application integrates advanced APIs for real-time data retrieval and processing. Visual elements like interactive charts and weather icons enhance data interpretation, while features such as dynamic search and highlight sections simplify navigation. This project is designed to address the limitations of existing weather tracking systems by offering a modern, user-centric solution that promotes climate awareness and supports informed decision-making in an increasingly unpredictable environment.

Code:

//index.html

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <script async src="https://www.googletagmanager.com/gtag/js?id=G-GQRS2F1JVF"></script>
```

```
  <script>
```

```
    window.dataLayer = window.dataLayer || [];
```

```
    function gtag(){dataLayer.push(arguments);}
```

```
<link rel="preconnect" href="https://fonts.googleapis.com">
```

```
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
```

```
<link  
href="https://fonts.googleapis.com/css2?family=Nunito+Sans:wght@400;600&display=swap  
" rel="stylesheet">
```

```
<link rel="stylesheet" href="assest/css/style.css">
```

```
<script type="module" src="./assest/js/app.js"></script>
```

```
<script type="module" src="./assest/js/route.js"></script>
```

```
</head>
```

```
<body>
```

```
<header class="header">
```

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

```
<a href="" class="logo">
```

```
<!---->
```

```
<h2>Weather Tracker Website</h2>
```

```
</a>
```

```
<div class="search-view" data-search-view>
```

```
<div class="search-wrapper">
```

```
<input type="search" name="search" class="search-field" placeholder="search  
city ..." autocomplete="off" data-search-field>
```

```
<span class="m-icon leading-icon">search</span>
```

```
<button class="icon-btn leading-icon has-state" aria-label="close search" data-  
search-toggler>
```

```
<span class="m-icon">arrow_back</span>
```

```
        </button>

    </div>

    <div class="search-result" data-search-result>

    </div>

</div>

<div class="header-actions">

    <button class="icon-btn has-state" aria-label="open search" data-search-toggler>

        <span class="m-icon">search</span>

    </button>

    <a href="#/current-location" class="btn-primary has-state" data-current-location-
btn>

        <span class="m-icon">my_location</span>

        <span class="span">Current Location</span>

    </a>

</div>

</div>

</header>

<main>

    <article class="container" data-container>

        <div class="content-left">

            <!-- current weater -->

            <section class="section current-weather" aria-label="current weather" data-current-
weather></section>

            <!-- forecast -->

            <section class="section forecast" aria-labelledby="forecast-label" data-5-day-
forecast></section>
```

```
</div>

<div class="content-right">

  <!-- highlight -->

  <section class="section highlights" aria-labelledby="highlights-label" data-
highlights></section>

  <section class="section hourly-forecast" aria-label="hourly forecast" data-hourly-
forecast></section>

</div>

<div class="loading" data-loading>

</div>

</article>

</main>

<!-- 404 Section -->

<section class="error-content" data-error-content>

  <h2 class="heading">404</h2>

  <p class="body-1">Page not found!</p>

  <a href="" class="btn-primary">

    <span class="span">Go Home</span>

  </a>

</section>

</body>

</html>
```

```
//app.js
```

```
'use strict';
```

```
import { fetchData,url } from "./api.js";
```

```

import * as module from "./module.js";

/**
 *
 * @param {NodeList} elements Elements node array
 * @param {String} eventType Event Type e.g: "click","mouseover"
 * @param {Function} callback callback function
 */

const addEventOnElements=(elements, eventType, callback)=>{
    for(const element of elements)
        element.addEventListener(eventType,callback);
}

const searchView = document.querySelector("[data-search-view]");
const searchTogglers = document.querySelectorAll("[data-search-toggler]");
const toggleSearch=()=>{
    searchView.classList.toggle("active");
}

addEventOnElements(searchTogglers,"click",toggleSearch);

// search integration

const searchField = document.querySelector("[data-search-field]");
const searchResult = document.querySelector("[data-search-result]");

let searchTimeOut =null;

let searchTimeOutDuration = 500;

```

```

searchField.addEventListener("input",()=>{

    searchTimeOut ?? clearTimeout(searchTimeOut);

    if(!searchField.value){

        searchResult.classList.remove("active");

        searchResult.innerHTML="";

        searchField.classList.remove("searching");

    }

    else{

        searchField.classList.add("searching");

    }

    if(searchField.value){

        clearTimeout(searchTimeOut)

        searchTimeOut=setTimeout(()=>{

            fetchData(url.geo(searchField.value),(locations)=>{

                searchField.classList.remove("searching");

                searchResult.classList.add("active");

                searchResult.innerHTML=`

                    <ul class="view-list" data-search-list></ul>

                `;

                const items=[];

                for (const {name, lat, lon, country, state} of locations){

                    const searchItem =document.createElement("li");

                    searchItem.classList.add("view-item");

                    searchItem.innerHTML=`

```

```

        <span class="m-icon">location_on</span>

        <div>

            <p class="item-title">${name}</p>

            <p class="label-2 item-subtitle">${state||""} ${country}</p>

        </div>

        <a href="#/weather?lat=${lat}&lon=${lon}" class="item-link has-state" aria-
label="${name} weather" data-search-toggler></a>

`;

searchResult.querySelector("[data-search-list]").appendChild(searchItem);

items.push(searchItem.querySelector("[data-search-toggler]"))

}

addEventOnElements(items,"click",()=>{

    toggleSearch();

    searchResult.classList.remove("active")

})

});

},searchTimeOutDuration);

}

});

```

```

const container = document.querySelector("[data-container]");

const loading = document.querySelector("[data-loading]");

const currentLocationBtn = document.querySelector("[data-current-location-btn]");

const errorContent = document.querySelector("[data-error-content]")

```

```

export const updateWeather = (lat,lon)=>{

  loading.style.display="grid";

  //container.style.overflowY="hidden";

  container.classList.remove("fade-in");

  errorContent.style.display="none";


  const currentWeatherSection =document.querySelector("[data-current-weather]");

  const highlightSection =document.querySelector("[data-highlights]");

  const hourlySection =document.querySelector("[data-hourly-forecast]");

  const forecastSection =document.querySelector("[data-5-day-forecast]");


  currentWeatherSection.innerHTML=""

  highlightSection.innerHTML=""

  hourlySection.innerHTML=""

  forecastSection.innerHTML=""


  if(window.location.hash == "#/current-location")

    currentLocationBtn.setAttribute("disabled","");

  else

    currentLocationBtn.removeAttribute("disabled");


  //CURRENT WEATHER


  fetchData(url.currentWeather(lat,lon),(currentWeather)=>{

    const{

```



```

    weather,

    dt: dateUnix,

    sys: { sunrise: sunriseUnixUTC, sunset: sunsetUnixUTC },

    main: { temp, feels_like, pressure, humidity },

    visibility,

    timezone

} = currentWeather;

const [{ description, icon }] = weather;

const card = document.createElement("div");

card.classList.add("card", "card-lg", "current-weather-card");

card.innerHTML=`

    <h2 class="title-2 card-title">Now</h2>

    <div class="weapper">

        <p class="heading">${parseInt(temp)}&deg;<sup>c</sup></p>

    </div>

    <p class="body-3">${description}</p>

    <ul class="meta-list">

        <li class="meta-item">

            <span class="m-icon">calendar_today</span>

            <p class="title-3 meta-text">${module.getDate(dateUnix,timezone)}</p>

        </li>

        <li class="meta-item">

            <span class="m-icon">location_on</span>

```

```
<p class="title-3 meta-text" data-location></p>
```

```
</li>
```

```
</ul>
```

```
,
```

```
fetchData(url.reverseGeo(lat,lon),([{ name, country }])=>{
```

```
  card.querySelector("[data-location]").innerHTML=`${name}, ${country}`;
```

```
})
```

```
currentWeatherSection.appendChild(card);
```

```
//today's highlights
```

```
fetchData(url.airPollution(lat,lon),(airPollution)=>{
```

```
  const[{
```

```
    main :{ aqi },
```

```
    components: { no2, o3, so2, pm2_5 }
```

```
  }]=airPollution.list;
```

```
const card=document.createElement("div");
```

```
card.classList.add("card","card-lg");
```

```
card.innerHTML=`
```

```
<h2 class="title-2" id="highlights-lable">Today Highlights</h2>
```

```
<div class="highlight-list">
```

```
  <div class="card card-sm highlight-card one">
```

```
    <h3 class="title-3">Air Quality Index</h3>
```

```
    <div class="wrapper">
```

air

<ul class="card-list">

<li class="card-item">

<p class="title-1">\${pm2_5.toPrecision(3)}</p>

<p class="label-1">PM_{2.5}</p>

<li class="card-item">

<p class="title-1">\${so2.toPrecision(3)}</p>

<p class="label-1">SO₂</p>

<li class="card-item">

<p class="title-1">\${no2.toPrecision(3)}</p>

<p class="label-1">No₂</p>

<li class="card-item">

<p class="title-1">\${o3.toPrecision(3)}</p>

<p class="label-1">O₃</p>

</div>

<span class="badge aqi-\${aqi} lable-\${aqi}"
title="\${module.aqiText[aqi].message}">

 \${module.aqiText[aqi].level}

</div>

```
<div class="card card-sm highlight-card two">

  <h3 class="title-3">Sunrise & Sunset</h3>

  <div class="wrapper">

    <div class="card-list">

      <div class="card-item">

        <span class="m-icon">clear_day</span>

        <div class="lable-1">

          <p class="lable-1">Sunrise</p>

          <p class="title-
1">${module.getTime(sunriseUnixUTC,timezone)}</p>

        </div>

      </div>

      <div class="card-item">

        <span class="m-icon">clear_night</span>

        <div class="lable-1">

          <p class="lable">Sunset</p>

          <p class="title-
1">${module.getTime(sunsetUnixUTC,timezone)}</p>

        </div>

      </div>

    </div>

  </div>

</div>

<div class="card card-sm highlight-card">

  <h3 class="title-3">Humidity</h3>

  <div class="wrapper">
```

humidity_percentage

<p class="title-1">\${humidity}_%</p>

</div>

</div>

<div class="card card-sm highlight-card">

<h3 class="title-3">Pressure</h3>

<div class="wrapper">

airwave

<p class="title-1">\${pressure} <sub>hba</sup></p>

</div>

</div>

<div class="card card-sm highlight-card">

<h3 class="title-3">Visibility</h3>

<div class="wrapper">

visibility

<p class="title-1">\${visibility/1000} _{KM}</p>

</div>

</div>

<div class="card card-sm highlight-card">

<h3 class="title-3">Feels Like</h3>

<div class="wrapper">

thermostat

<p class="title-1">\${parseInt(feels_like)}°^c</p>

</div>

</div>

```

</div>

`;

highlightSection.appendChild(card)

}))

//24H forecast

fetchData(url.forecast(lat,lon),(forecast)=>{

  const{

    list: forecastList,

    city:{timezone}

  } = forecast;

  hourlySection.innerHTML=`

    <h2 class="title-2">Today at</h2>

    <div class="slider-container">

      <ul class="slider-list" data-temp></ul>

      <ul class="slider-list" data-wind></ul>

    </div>

  `;

  for (const[index,data] of forecastList.entries()){

    if(index>7)

      break

    const{

      dt: dateTimeUnix,

      main: {temp},

```

```

        weather,

        wind: {deg:windDirection, speed:windSpeed}

    }=data;

    const[{icon,description}]=weather;

    const tempLi=document.createElement("li");

    tempLi.classList.add("slider-item");

    tempLi.innerHTML=`

        <div class="card card-sm slider-card">

            <p class="body-3">${module.getTime(dateTimeUnix,timezone)}</p>

            <p class="body-3">${temp}&deg;</p>

        </div>

    `;

    hourlySection.querySelector("[data-temp]").appendChild(tempLi);

    const windLi = document.createElement("li");

    windLi.classList.add("slider-item");

    windLi.innerHTML=`

        <div class="card card-sm slider-card">

            <p class="body-3">${module.getTime(dateTimeUnix,timezone)}</p>

            <p class="body-3">${parseInt(module.mps_to_kmh(windSpeed)) }Km/h</p>

        </div>

    `;

```

```
    hourlySection.querySelector("[data-wind]").appendChild(windLi);  
  }  
}
```

```
//5 day forecast
```

```
forecastSection.innerHTML=`
```

```
  <h2 class="title-2" id="forecast-label">5 Days Forecast</h2>
```

```
  <div class="card card-lg forecast-card">
```

```
    <ul data-forecast-list></ul>
```

```
  </div>
```

```
`;  
`;
```

```
for(let i = 7, len = forecastList.length; i < len; i += 8){
```

```
  const{
```

```
    main:{temp_max},
```

```
    weather,
```

```
    dt_txt
```

```
  }=forecastList[i];
```

```
  const [{icon,description}]=weather;
```

```
  const date = new Date(dt_txt);
```

```
  const li = document.createElement("li");
```

```
  li.classList.add("card-item");
```

```
  li.innerHTML=`
```

```
    <div class="icon-wrapper">
```

```
      
```



```

        <span class="span">

        <p class="title-2">${parseInt(temp_max)}&deg;</p>

        </span>

    </div>

    <p class="label-1">${date.getDate()}
    ${module.monthNames[date.getMonth()]}</p>

    <p class="label-1">${module.weekDayNames[date.getUTCDay()]}</p>
    `;

    forecastSection.querySelector("[data-forecast-list]").appendChild(li)

    }

    loading.style.display="none";

    container.classList.add("fade-in");

    });

    });

    }

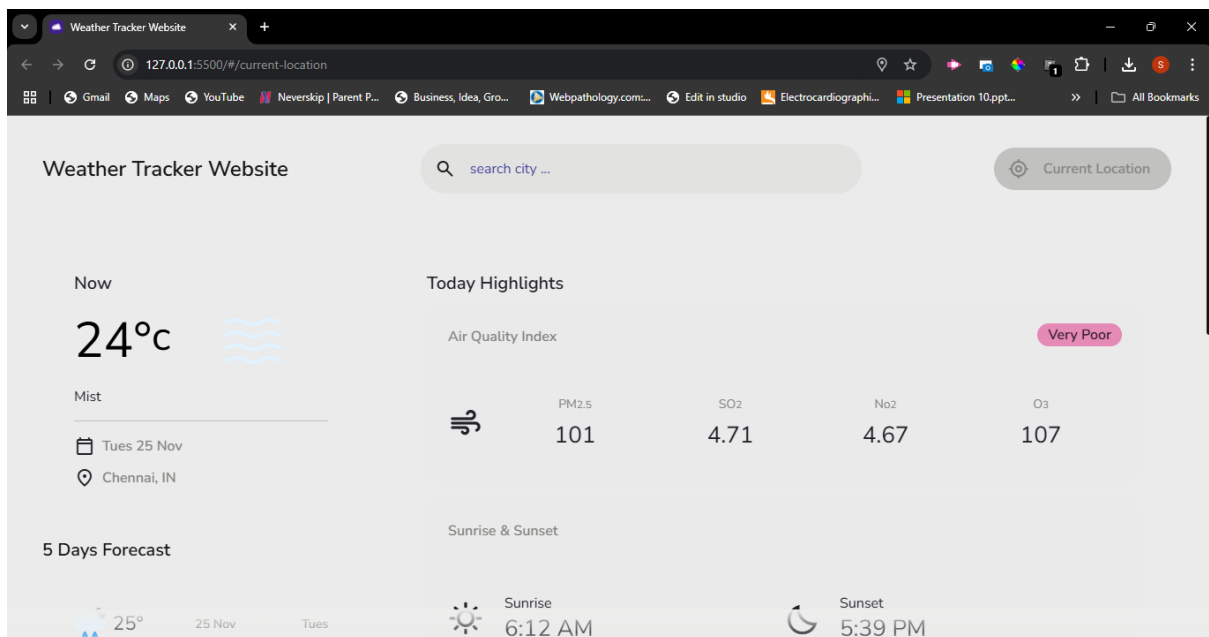
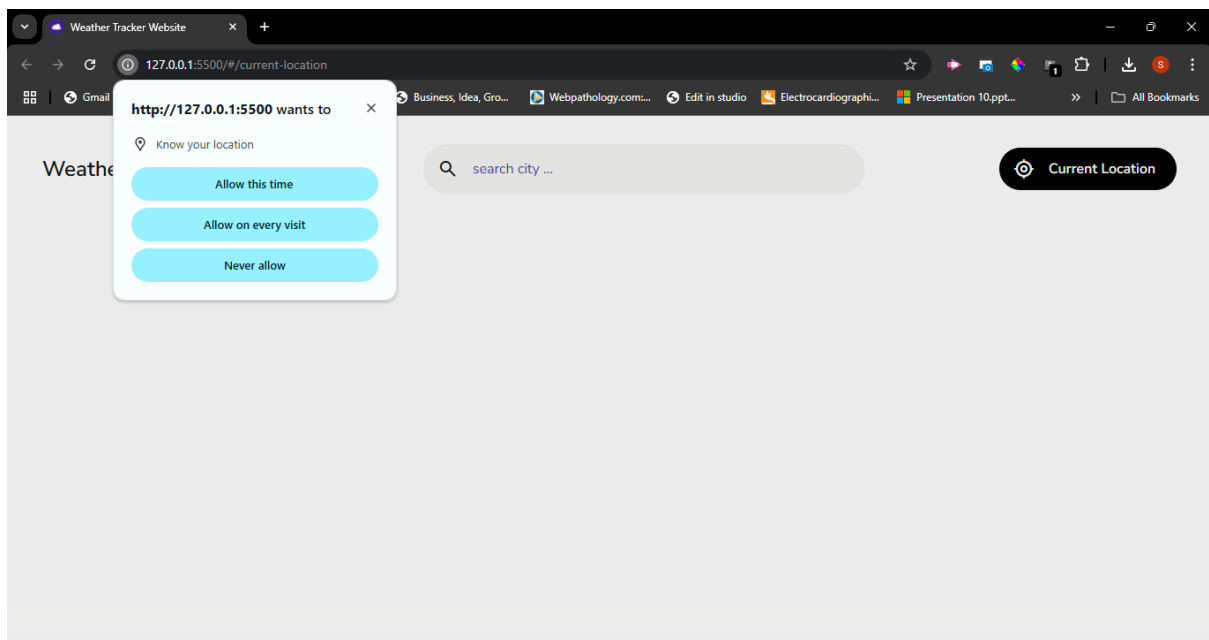
export const error404=()=>{

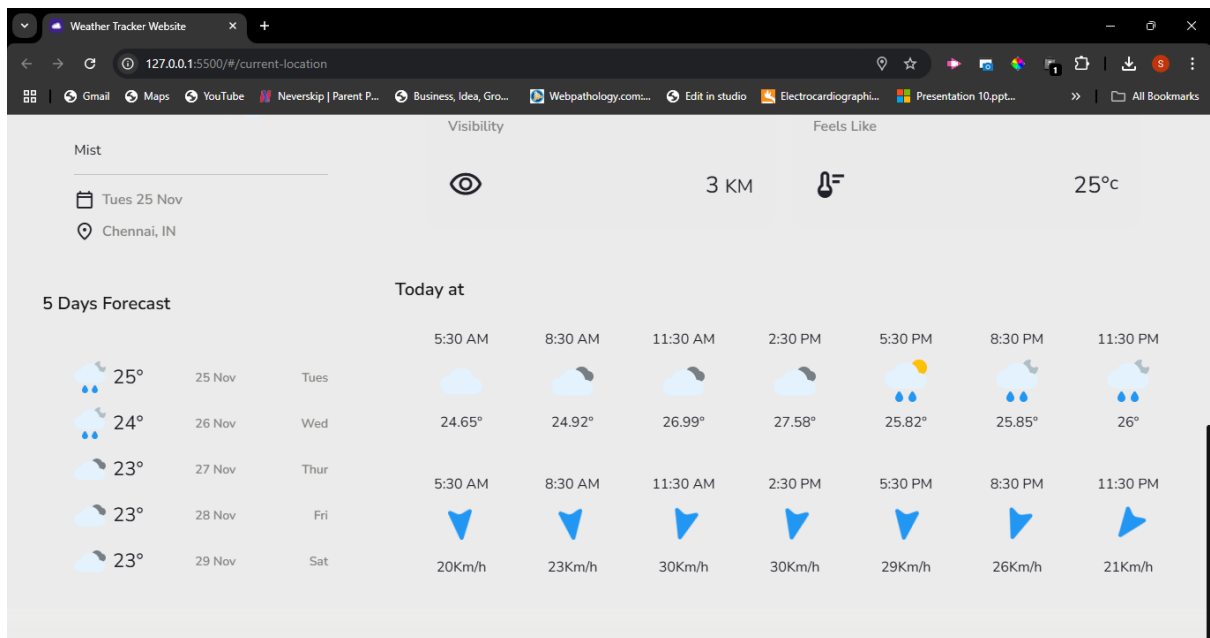
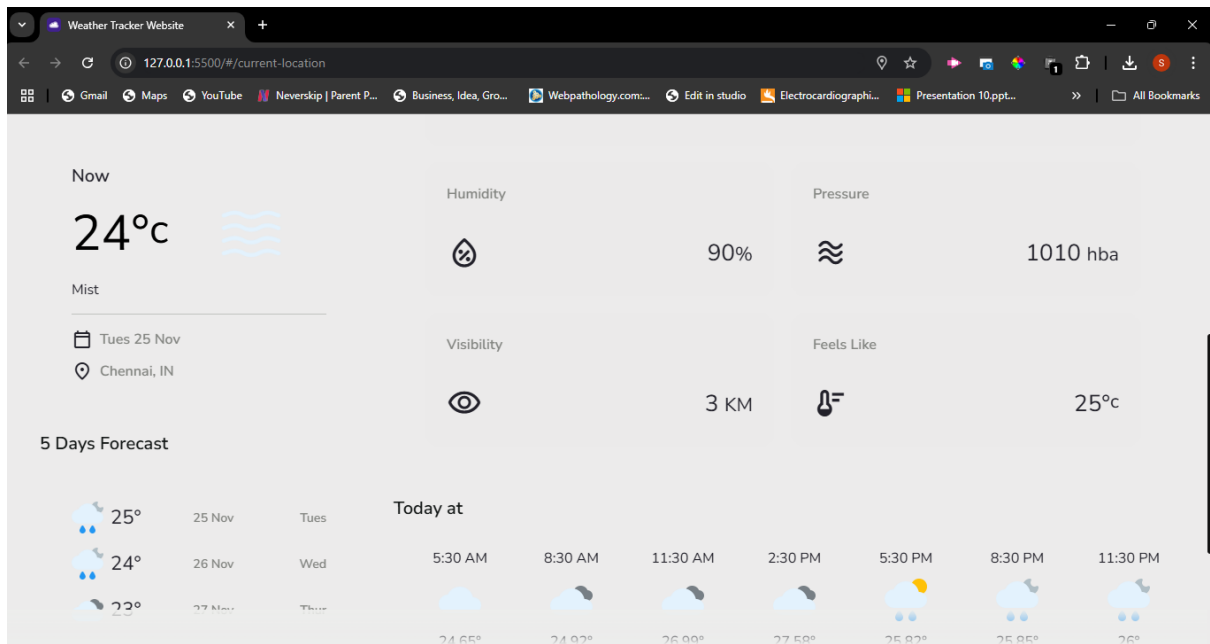
    errorContent.style.display="flex"

    };

```

Output Screenshots:





Project Conclusion

The Weather Tracking Application successfully meets the growing need for reliable, real-time weather information in an easy-to-use and accessible format. By integrating advanced APIs and implementing a responsive, intuitive interface, the application ensures users have accurate weather data at their fingertips, whether for everyday planning or more specific, climate-dependent decisions. The dynamic search feature, location-based tracking, and detailed 5-day forecasts address the limitations of traditional weather systems, offering an enhanced user experience. Through this project, we have demonstrated the importance of combining modern technology with user-centric design to provide comprehensive weather insights. The application not only improves climate awareness but also empowers individuals to make informed decisions based on up-to-date meteorological data. As the project evolves, further enhancements such as additional weather metrics, localization, and real-time notifications could make it an even more essential tool for users worldwide.