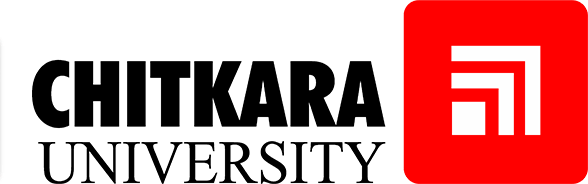
**Front End Engineering-II**

Project Report

Semester-IV

(Batch-2022)

**WEATHER APP**



**Supervised By:                                                                              Submitted By:**

Dr. Raveesh Samkaria Shivam Mainro

2210990825

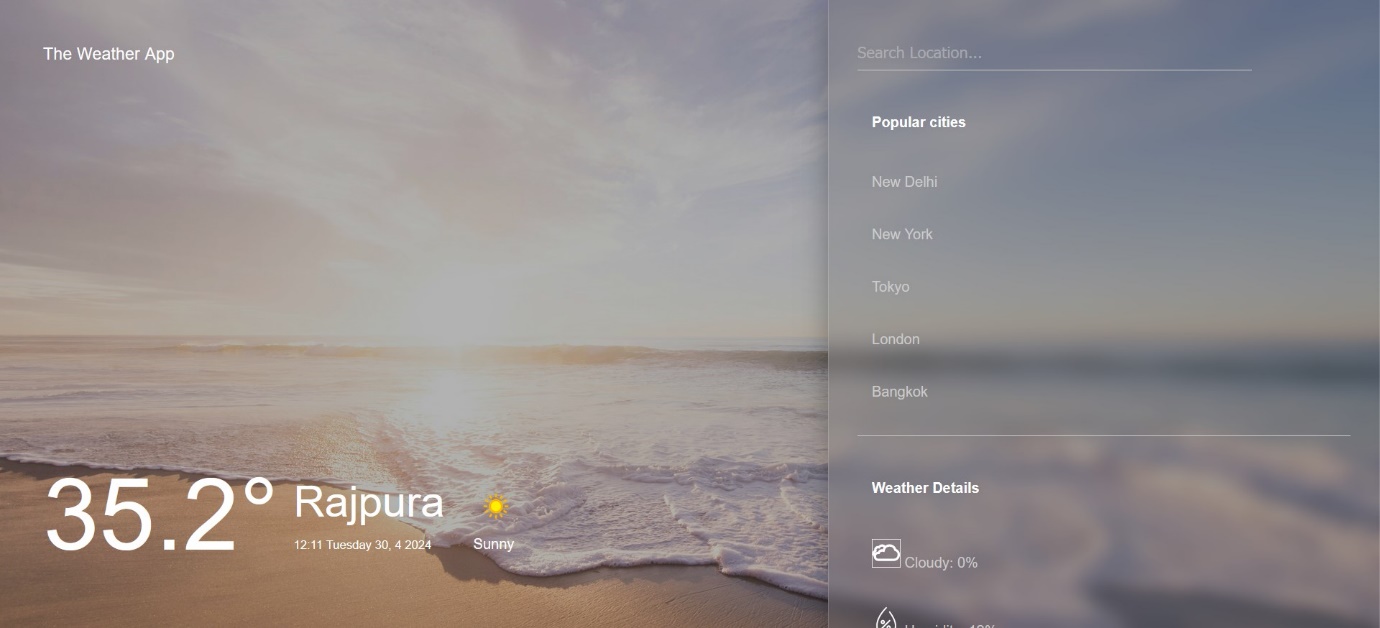
G-12

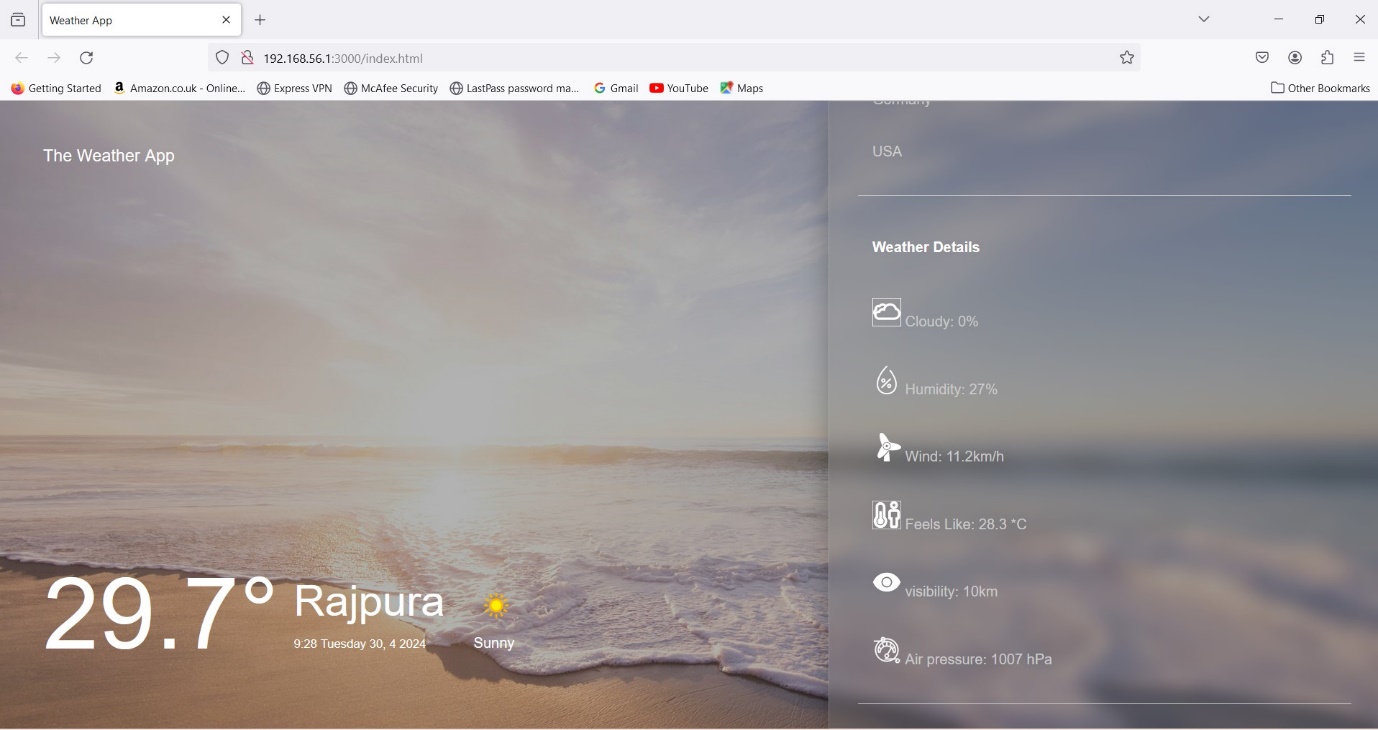
**Department of Computer Science and Engineering Chit- kara University Institute of Engineering & Technology,**

**Chitkara University, Punjab**

**Abstract**

This user-friendly weather app “The Weather App” empowers you to stay informed about current weather conditions with ease. The app shows a responsive and visually appealing interface that adapts seamlessly to any device, from desktops to smartphones.





**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Title** | **Page Number(s)** |
| 1 | Introduction | 4 |
| 2 | Problem Statement | 5 |
| 3 | Software Requirements | 5 |
| 4 | Proposed Design | 6-13 |
| 5 | Results | 14-17 |
| 6 | References | 18 |

1. **Introduction**

The Weather App web application offers a convenient yet efficient platform for checking weather based on current location. In today's digital age, where convenience and accessibility are paramount, such tools provide valuable assistance in everyday tasks. This introduction provides an overview of the background, objectives, and significance of the “The Weather App”.

1. **Background:**

As technology continues to permeate various aspects of our lives, the need for quick and easy- to-use utilities becomes increasingly apparent. The “Weather App” web app addresses a fundamental need i.e. the ability to check accurate weather anytime needed and anywhere needed . With the proliferation of web-based applications, users expect seamless experiences that deliver results promptly and accurately.

1. **Objectives:**

The primary objective of “The Weather App” is to provide a user-friendly interface for individuals to input city name and obtain the weather instantly. This application aims to simplify the process, eliminating the need for users to browse through internet or rely on external tools. Additionally, “The Weather App” web app strives to ensure accuracy.

1. **Significance:**

The significance of “The Weather App” web app lies in its ability to streamline a common task that individuals encounter regularly. Whether for personal use, professional purposes, or administrative requirements, knowing the weather accurately is essential. By offering a convenient and reliable solution, the “Weather App” web app enhances efficiency and productivity, saving users time and effort.

1. **Problem Statement**

Existing weather forecast web apps are cluttered with useless information and prone to errors, lacking accuracy and user-friendliness. Current online applications often overlook various factors compromising accuracy. There's a need for a precise, intuitive, and accessible Weather app that ensures accuracy while considering all relevant factors, catering to users' diverse needs across different platforms and devices. Additionally, the lack of attention to detail in existing solutions hinders their reliability and usability, resulting in frustration for users seeking quick and accurate weather information.

1. **Software Requirements**

**a. Integrated Development Environment (IDE):**

* + Visual Studio Code (VS Code) for code editing and project management.

**b.Frontend Technologies:**

* + HTML: Markup language for structuring the web application.
  + CSS: Styling language for enhancing the presentation and layout.
  + JavaScript (JS): Programming language for implementing interactive features and logic and making the API call.

**c. Version Control:**

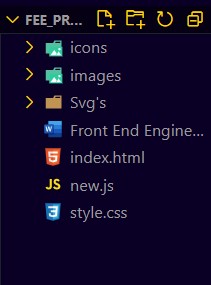
* + Git: Distributed version control system for tracking changes in the project codebase.

1. **Proposed Design**

* **User Interface Design:** Utilize CSS Media-Query for a responsive.
* **Frontend Development:** Develop using HTML, CSS, and JavaScript. Utilize HTML5 semantics, CSS for styling, and JavaScript for dynamic UI updates.
* **User Experience Optimization:** Focus on real-time feedback, interactive elements, and cross-browser compatibility. Ensure responsiveness for varied devices.
* **Documentation and Deployment:** Provide detailed documentation. Deploy on web server with domain. Maintain and update documentation regularly.
* **Availability for the Community:** This project is available on Github on my Profile in a public repository so that people can check it out and can learn.

1. **File Structure**

Ensuring proper file and folder structure to maintain consistent file paths and clean structure.



1. **HTML Code Structure**

These screenshots present the HTML code for our” Weather App” web app project, revealing the layout and content of our web pages in a code format.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Weather App </title>

    <link rel="stylesheet" href="style.css">

</head>

<body>

    <div class="weather-app">

        <div class="container">

            <h3 class="brand">The Weather App</h3>

            <div>

                <h1 class="temp">16&#176;</h1>

                <div class="city-time">

                    <h1 class="name">Rajpura</h1>

                    <small>

                        <span class="time">06:09</span>

                        <span class="date">

                            Monday Sep 19

                        </span>

                    </small>

                </div>

                <div class="weather">

                    <img src="./icons/day/113.png" class="icon" alt="icon" width="50" height="50" />

                    <span class="condition">

                </div>

            </div>

        </div>

        <div class="panel">

            <form id="locationInput">

                <input type="text" class="search" placeholder="Search Location..." />

            </form>

            <ul class="cities">

                <h4>Popular cities</h4>

                <li class="city">New Delhi</li>

                <li class="city">New York</li>

                <li class="city">Tokyo</li>

                <li class="city">London</li>

                <li class="city">Bangkok</li>

            </ul>

            <ul class="details">

                <h4>Weather Details</h4>

                <li>

                    <img src="/Svg's/cloudy.svg" class="logo-2"  alt=""  >

                    <span>Cloudy:</span>

                    <span class="cloud">89%</span>

                </li>

                <li>

                    <img src="/Svg's/humidity.svg" class="logo-2"  alt=""  >

                    <span>Humidity:</span>

                    <span class="humidity">64%</span>

                </li>

                <li>

                    <img src="/Svg's/wind.svg" class="logo-2"  alt=""  >

                    <span>Wind:</span>

                    <span class="wind">8km/h</span>

                </li>

                <li>

                    <img src="/Svg's/feels\_like.svg" class="logo-2"  alt=""  >

                    <span>Feels Like:</span>

                    <span class="feels\_like">20 \*C</span>

                </li>

                <li>

                    <img src="/Svg's/visibility.svg" class="logo-2"  alt=""  >

                    <span>visibility:</span>

                    <span class="visibility">16km</span>

                </li>

                <li>

                    <img src="/Svg's/air-pressure.svg" class="logo-2"  alt=""  >

                    <span>Air pressure:</span>

                    <span class="air\_pressure">1006hPa</span>

                </li>

            </ul>

        </div>

    </div>

    <script src="new.js"></script>

</body>

</html>

1. **CSS Code Structure**

This screenshot exhibits the CSS code for our “Weather App” web app project, illustrating the styling and design elements implemented across our web pages.

\* {

    box-sizing: border-box;

    list-style: none;

}

body {

    margin: 0;

    font-family: 'Rubik', sans-serif;

    background: #111;

}

h1,

h3 {

    font-weight: 400;

}

.weather-app {

    min-height: 100vh;

    background-image: url(./images/day/cloudy.jpg );

    background-position: center;

    background-repeat: no-repeat;

    background-size: cover;

    color: #fff;

    position: relative;

    transition: 500ms;

    opacity: 1;

}

.weather-app::before {

    content: "";

    position: absolute;

    top: 0;

    left: 0;

    width: 100%;

    height: 100%;

    background: rgba(0, 0, 0, 0.3);

    z-index: 0;

}

.container {

    position: absolute;

    top: 0;

    left: 0;

    width: 100%;

    height: 100%;

    display: flex;

    justify-content: space-between;

    align-items: flex-start;

    flex-direction: column;

    padding: 2em 3em 4em 3em;

}

.container>div {

    display: flex;

    justify-content: center;

    align-items: center;

}

.city-time,

.temp,

.weather {

    margin: 0 1em;

}

.city-time h1 {

    margin: 0;

    margin-bottom: 0.2em;

    font-size: 3em;

}

.temp {

    font-size: 7em;

    margin: 0;

}

.weather img {

    display: block;

    margin: 0.5em 0;

}

.panel {

    position: absolute;

    width: 40%;

    height: 100%;

    top: 0;

    right: 0;

    background:

        rgba(110, 110, 110, 0.25);

    box-shadow:

        0 8px 32px 0 rgba(0, 0, 0, 0.3);

    backdrop-filter: blur(10px);

    -webkit-backdrop-filter: blur(10px);

    border: 1px solid rgba(255, 255, 255, 0.18);

    z-index: 1;

    padding: 3em 2em;

    overflow-Y: scroll;

}

.panel form {

    margin-bottom: 3em;

}

.submit {

    position: absolute;

    top: 0;

    right: 0;

    padding: 1.5em;

    margin: 0;

    border: none;

    outline: none;

    background: #fa6d1b;

    color: #fff;

    cursor: pointer;

    font-size: 1.2em;

    transition: 0.4s;

}

.submit:hover {

    background: #fff !important;

    color: #000;

}

.search {

    background: none;

    border: none;

    border-bottom: 1px #ccc solid;

    padding: 0 1em 0.5em 0;

    width: 80%;

    color: #fff;

    font-size: 1.1em;

}

.search:focus {

    outline: none;

}

.search::placeholder {

    color: #ccc;

}

.panel ul {

    padding: 00 1em 0;

    margin: 2em 0;

    border-bottom: 1px #ccc solid;

}

.panel ul li {

    color: #ccc;

    margin: 2.5em 0;

}

.panel ul h4 {

    margin: 3em 0;

}

.city {

    display: block;

    cursor: pointer;

}

.city:hover {

    color: #fff;

}

details li {

    display: flex;

    justify-content: space-between;

    align-items: center;

}

.logo-2{

    height: 2rem;

}

@media screen and (max-width: 800px) {

    .panel,

    .container {

        position: relative;

        width: 100%;

        top: initial;

    }

}

@media screen and (max-width: 500px) {

    html {

        font-size: 12px;

    }

}

@media screen and (max-height: 300px) {

    .weather-app {

        min-height: 40em;

    }

}

1. **JavaScript Code Structure**

This screenshot exhibits the JS code for our “Weather App” Web app project, illustrating the different functions, events and API Calls we trigger according to different scenarios.

const app = document.querySelector('.weather-app');

const temp = document.querySelector('.temp');

const dateOutput = document.querySelector('.date');

const timeOutput = document.querySelector('.time');

const conditionOutput = document.querySelector('.condition');

const nameOutput = document.querySelector('.name');

const icon = document.querySelector('.icon');

const cloudOutput = document.querySelector('.cloud');

const humidityOutput = document.querySelector('.humidity');

const windOutput = document.querySelector('.wind');

const feels\_likeOutput = document.querySelector('.feels\_like');

const visibilityOutput = document.querySelector('.visibility');

const air\_pressureOutput = document.querySelector('.air\_pressure');

const form = document.getElementById('locationInput');

const search = document.querySelector('.search');

const btn = document.querySelector('.submit');

const cities = document.querySelectorAll('.city');

let cityInput = "Rajpura";

cities.forEach((city) => {

    city.addEventListener('click', (e) => {

        cityInput = e.target.innerHTML;

        fetchWeatherData();

        app.style.opacity = "0";

    });

});

form.addEventListener('submit', (e) => {

    if (search.value.length == 0) {

        alert('Please type in a city name');

    } else {

        cityInput = search.value;

        fetchWeatherData();

        search.value = "";

        app.style.opacity = "0";

    }

    e.preventDefault();

});

function dayOfTheWeek(day, month, year) {

    const weekday = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

    return weekday[new Date(`${year}-${month}-${day}`).getDay()];

}

function fetchWeatherData() {

    fetch(`http://api.weatherapi.com/v1/current.json?key=cd13a49bf2af4f168cc14108242204&q=${cityInput}`)

        .then(response => response.json())

        .then(data => {

            console.log(data);

            temp.innerHTML = data.current.temp\_c + "&#176;";

            conditionOutput.innerHTML = data.current.condition.text;

            const date = data.location.localtime;

            const y = parseInt(date.substr(0, 4));

            const m = parseInt(date.substr(5, 2));

            const d = parseInt(date.substr(8, 2));

            console.log(date);

            const time = date.substr(11);

            dateOutput.innerHTML = `${dayOfTheWeek(d, m, y)} ${d}, ${m} ${y}`;

            timeOutput.innerHTML = time;

            nameOutput.innerHTML = data.location.name;

            const iconId = data.current.condition.icon.split('/').pop();

            icon.src = `./icons/${iconId}`;

            cloudOutput.innerHTML = data.current.cloud + "%";

            humidityOutput.innerHTML = data.current.humidity + "%";

            windOutput.innerHTML = data.current.wind\_kph + "km/h";

            feels\_likeOutput.innerHTML = data.current.feelslike\_c  + " \*C";

            visibilityOutput.innerHTML = data.current.vis\_km +"km";

            air\_pressureOutput.innerHTML = data.current.pressure\_mb + " hPa";

            let timeOfDay = "day";

            const code = data.current.condition.code;

            if (!data.current.is\_day) {

                timeOfDay = "night";

            }

            if (code == 1000) {

                app.style.backgroundImage = `url(./images/${timeOfDay}/clear.jpg)`;

                icon.src = `./icons/${timeOfDay}/113.png`;

                btn.style.background = "#e5ba92";

                if (timeOfDay == "night") {

                    btn.style.background = "#181e27";

                }

            } else if (code == 1003 || code == 1006 || code == 1009 || code == 1030 || code == 1069 || code == 1087 || code == 1135 || code == 1273 || code == 1276 || code == 1279 || code == 1282) {

                app.style.backgroundImage = `url(./images/${timeOfDay}/cloudy.jpg)`;

                icon.src = `./icons/${timeOfDay}/116.png`;

                btn.style.background = "#fa6d1b";

                if (timeOfDay == "night") {

                    btn.style.background = "#181e27";

                }

            } else if (code == 1063 || code == 1069 || code == 1072 || code == 1150 || code == 1153 || code == 1180 || code == 1183 || code == 1186 || code == 1189 || code == 1192 || code == 1195 || code == 1204 || code == 1207 || code == 1240 || code == 1243 || code == 1246 || code == 1249 || code == 1252) {

                app.style.backgroundImage = `url(./images/${timeOfDay}/rainy.jpg)`;

                icon.src = `./icons/${timeOfDay}/263.png`;

                btn.style.background = "#647d75";

                if (timeOfDay == "night") {

                    btn.style.background = "#325c80";

                }

            } else {

                app.style.backgroundImage = `url(./images/${timeOfDay}/snowy.jpg)`;

                icon.src = `./icons/${timeOfDay}/332.png`;

                btn.style.background = "#4d72aa";

                if (timeOfDay == "night") {

                    btn.style.background = "#1b1b1b";

                }

            }

            app.style.opacity = "1";

        })

        .catch(() => {

            app.style.opacity = "1";

        });

}

fetchWeatherData();

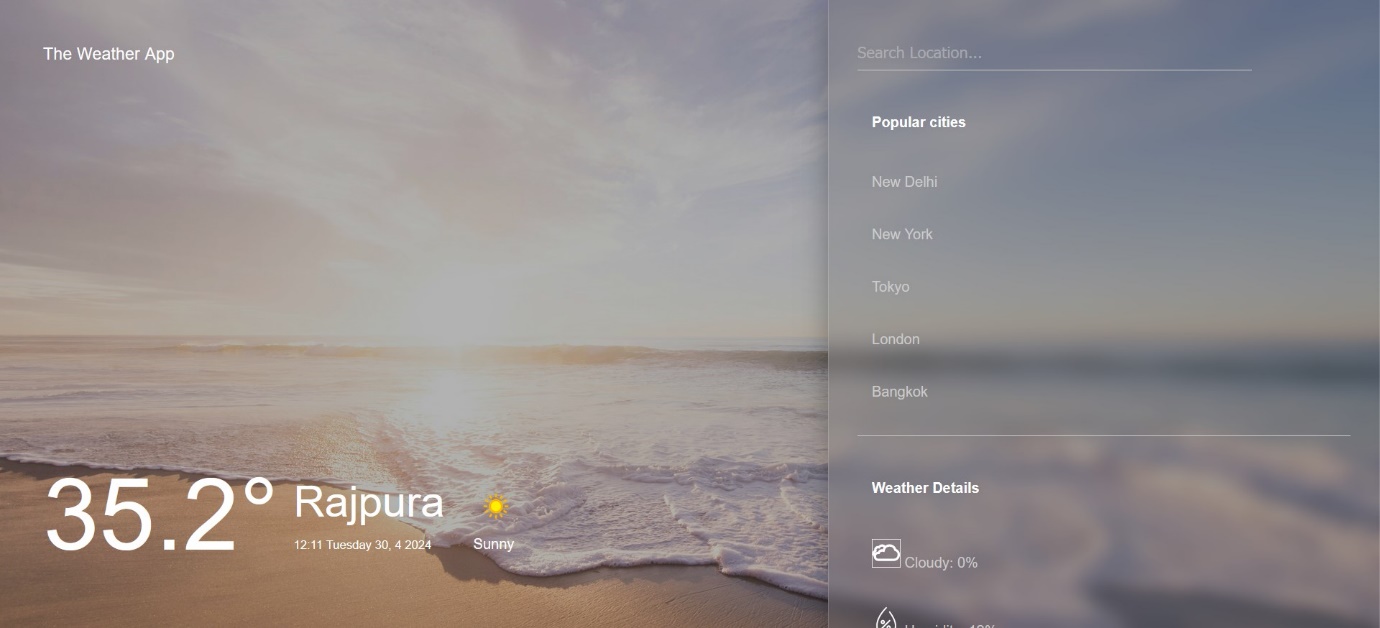
app.style.opacity = "1";

**Results**

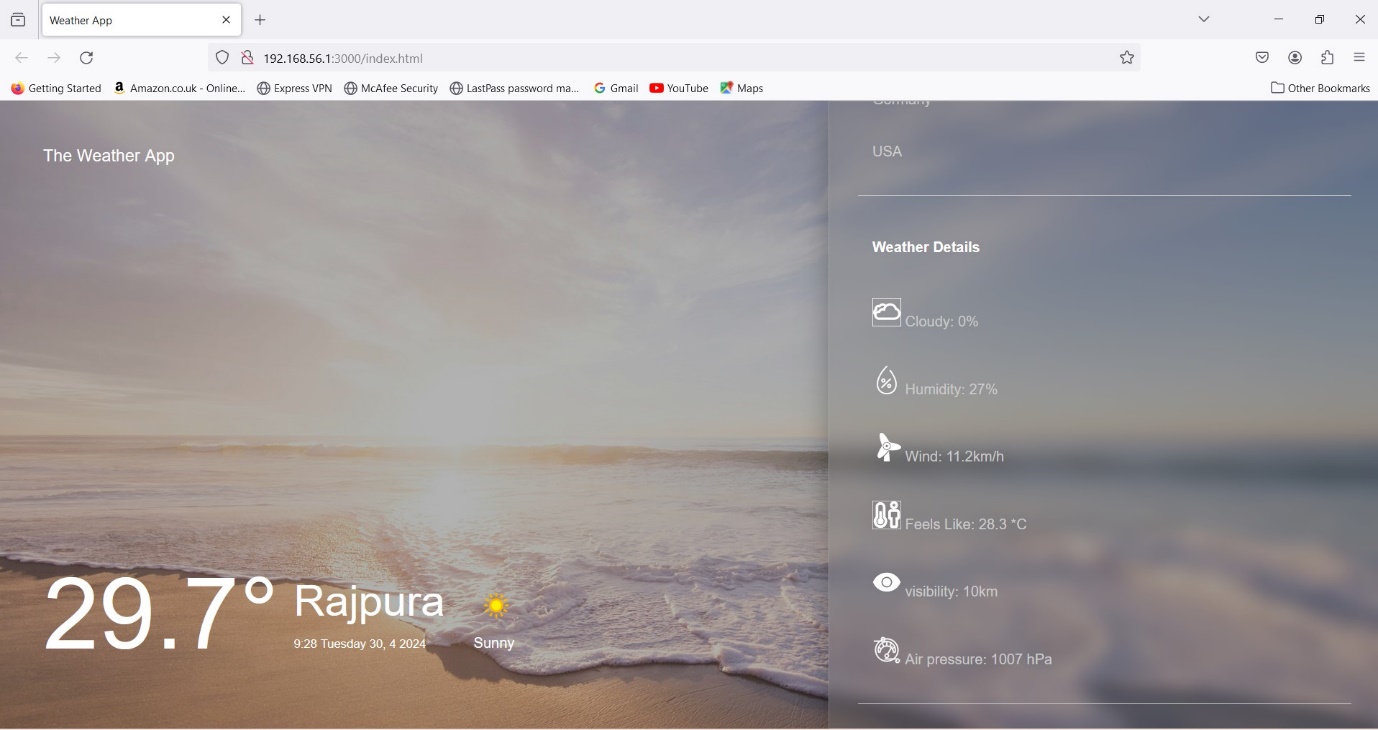
The “Weather App” application successfully fulfils its intended purpose of accurately showing the Weather in the entered Area. Through rigorous testing and user feedback, the following key outcomes have been achieved:

**GitHub Repository Link:**

**Project Screenshots for all scenarios:**



Static Scenario



Data Presenting Scenario

* **Accuracy:** The “Weather App” logic, implemented in JavaScript, accurately determines the weather in the entered Location.
* **User Experience:** The user interface design, Css media-query provides a seamless and visually appealing experience across different devices and screen sizes. Interactive elements and real-time feedback enhance usability and engagement.
* **Functionality:** Users can easily input city name through the intuitive interface, triggering the API call process with pressing Enter . The fetched Weather is displayed dynamically.
* **Documentation and Deployment:** Comprehensive documentation guides users through setup, usage, and troubleshooting. Deployment on a web server with domain access ensures public availability, while regular updates maintain relevance and accuracy.
* **Availability for the Community:** This project is available on Github on my Profile in a public repository so that people can check it out and can learn.

Overall, the “Weather App” application delivers on its objectives, providing users with a reliable, intuitive, and enjoyable tool for determining the weather with ease and accuracy.

**References**

* **HTML, CSS, and JavaScript Documentation:**
  + **Mozilla Developer Network (MDN) - HTML:** https://devel- oper.mozilla.org/en-US/docs/Web/HTML
  + **Mozilla Developer Network (MDN) - CSS:** https://developer.mozilla.org/en-

US/docs/Web/CSS

* **Mozilla Developer Network (MDN) - JavaScript:** https://devel- oper.mozilla.org/en-US/docs/Web/JavaScript
* **Frontend Development Tutorials and Articles:**
  + **CSS-Tricks:** <https://css-tricks.com/>
  + **JavaScript.info:** <https://javascript.info/>
* **API Reference:**
  + - **Link:**  <https://www.weatherapi.com/my/>