A FIELD PROJECT REPORT ON

**DEVELOPING A WEATHER APPLICATION**

Submitted in partial fulfilment of the requirements for the award of the degree

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE OF ENGINEERING**

Submitted by

|  |  |
| --- | --- |
| P.NARAYANA REDDY - 231FA04065  M.ASHOK KUMAR -231FA04080  G.SASIDHAR -231FA04369 |  |

K.GOWTHAM -231FA04949



Department of COMPUTER SCIENCE AND ENGINEERING

Vignan’s Foundation for Science, Technology and Research (Deemed to be University) Vadlamudi, Guntur, Andhra Pradesh-522213, India

MARCH– 2025



**CERTIFICATE**

This is to certify that the field project entitled

“DEVELOPING A WEATHER APPLICATION” being submitted by (P.NARAYANA REDDY – 231FA04065), (M.ASHOK KUMAR – 231FA04080), (G.SASIDHAR – 231FA04369), and (K.GOWTHAM – 231FA04949) in partial fulfilment of Bachelor of Technology in the Department of COMPUTER SCIENCE AND ENGINEERING, Vignan’s Foundation For Science Technology & Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India, is a bonafide work carried out by them under my guidance and supervision.

|  |  |
| --- | --- |
| **Head of the Department** | **Guide** |



**DECLARATION**

We hereby declare that our project work described in the field project titled “DEVELOPING A WEATHER APPLICATION” which is being submitted by us for the partial fulfilment in the department of COMPUTER SCIENCE AND ENGINEERING, Vignan’s Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, and the result of investigations are carried out by us under the guidance of (DR.R.PRATHAP KUMAR)

P.NARAYANA REDDY - 231FA04065 Signature : Narayana reddy

M.ASHOK KUMAR - 231FA04080 Signature : Ashok kumar

G.SASIDHAR - 231FA04369 Signature : Sasidhar

K.GOWTHAM - 231FA04949 Signature : Gowtham

**Contents**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **DESCRIPTION** | **PAGE NO.** |
| **1.** | **Introduction** | **5-6** |
| **2.** | **System Requirements** | **7** |
| **3.** | **Implementation** | **8-14** |
| **4.** | **Results** | **15-18** |
| **5.** | **Conclusion** | **19** |
| **6.** | **References** | **20** |

**1.INTRODUCTION**

Weather plays a crucial role in daily activities such as transportation, agriculture, outdoor planning, and disaster management. With the increasing unpredictability of climate patterns, there is a need for an efficient and accessible weather forecasting system that provides real-time updates.

The Weather Check Web App is developed to address this need by offering instant weather updates using the Open Weather API. The application is built with HTML, CSS, and JavaScript and provides users with temperature, weather conditions, and wind speed in an easy-to-use interface.

**1.1Problem Definition:-**

Weather updates are crucial for planning various activities, but existing solutions present the following challenges:

Traditional methods (TV, radio) provide weather updates at fixed intervals and lack real-time updates.

Most weather applications require installation and consume storage space.

Some online weather services provide incomplete data unless users subscribe to premium plans.

The lack of an intuitive user interface (UI) makes weather data difficult to interpret for general users.

Objective

The objective of this project is to develop a web-based weather application that offers real-time weather updates, ensuring ease of access, security, and an enhanced user experience.

**1.2Existing System;-**

**Current Methods for Weather Forecasting:**

TV and Radio Forecasts – Provide scheduled updates but lack real-time data.

Mobile Weather Applications – Require installation, updates, and may include ads or subscription models.

Web-Based Weather Services – Some are not mobile-friendly and lack interactive features.

**1.3Proposed System;-**

The Weather Check Web App is designed to overcome the limitations of existing systems by offering:

Real-time weather updates from the OpenWeather API.

Secure login authentication to ensure restricted access.

Modern and responsive UI with dynamic visual effects.

Cross-platform compatibility, making it usable on both desktops and mobile devices.

Error handling mechanisms to notify users of incorrect inputs or connectivity issues.

**Key Features:**

Users can log in securely to access the application.

The application retrieves weather data based on user input.

The UI is designed with glassmorphism and dynamic backgrounds to enhance usability.

The app provides real-time error notifications for incorrect entries.

Works efficiently on both mobile and desktop platforms.

**1.4 Literature Review: -**

Real-time Weather Forecasting using APIs

Discusses how third-party APIs can be integrated to fetch live meteorological data efficiently.

UI/UX Trends in Web Applications

Highlights the importance of interactive design and responsiveness for better user engagement.

Security Measures in Web Applications

Covers various authentication techniques for securing user data in web-based systems.

Meteorological Data Processing and Challenges

Explores the accuracy and reliability of different weather forecasting techniques.

**2.SYSTEM REQURIMENTS**

**Software Requirements:**

Frontend: HTML, CSS, JavaScript

Backend: OpenWeather API for weather data retrieval

Development Tools: VS Code, Sublime Text, Notepad++

Supported Browsers: Chrome, Firefox, Edge

Hardware Requirements:

Internet Connection: Required for fetching live data

Minimum System Requirements: 4GB RAM, Dual-core processor

Storage: Adequate space for project files and dependencies

**Problem Design:-**

Problem Design for Weather Check Web Application

The Problem Design phase involves structuring the system architecture, flow, and components to ensure a functional and efficient weather application. Below are the key aspects of the problem design:

System Architecture

The Weather Check Web App follows a three-layer architecture:

Presentation Layer (Frontend) → User Interface (HTML, CSS, JavaScript).  
Business Logic Layer (Backend Processing) → Fetching data from OpenWeather API.  
 Data Layer (API Integration) → Retrieving real-time weather details.

**Implementation:-**

**1. User Authentication:**

Secure login system ensures that only authorized users can access the application.

Users must enter a valid username and password to proceed.

**2. Weather Data Retrieval:**

Users input a city name in the search bar.

The app sends a request to the **OpenWeather API** to retrieve the latest weather conditions.

The system dynamically displays **temperature, weather conditions, wind speed, and the respective country’s flag**.

**3. User Interface & Design:**

Implements **glassmorphism UI with dynamic backgrounds** for an engaging experience.

Features a **responsive layout**, ensuring compatibility with both mobile and desktop devices.

Weather details are displayed in a **clean and easy-to-read format**.

**4. Error Handling:**

Users receive notifications for **invalid city names or network issues**.

Displays meaningful error messages for a better user experience.

**Sample code:-**

**HTML:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Weather Check</title>

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

</head>

<body onload="updateDate()">

    <div id="login-container">

        <h2>Get Weather - Login</h2>

        <input type="text" id="username" placeholder="Enter Username">

        <input type="password" id="password" placeholder="Enter Password">

        <button onclick="login()">Login</button>

    </div>

    <div id="weather-app" style="display:none;">

        <div class="container">

            <div class="weather-side">

                <div class="date-container">

                    <h2 class="date-dayname">Loading...</h2>

                </div>

                <div class="weather-container">

                    <h1 class="weather-city">--</h1>  <!-- City, Country & Flag -->

                    <h1 class="weather-temp">--°C</h1>  <!-- Temperature -->

                    <h3 class="weather-desc">--</h3>  <!-- Weather Condition -->

                    <h3 class="weather-wind">Wind: -- km/h</h3>  <!-- Wind Speed -->

                </div>

            </div>

            <div class="location-container">

                <input type="text" id="location-input" placeholder="Enter Location">

                <button onclick="updateWeather()">Get Weather</button>

            </div>

        </div>

    </div>

    <script src="script.js"></script>

</body>

</html>

**CSS:-**

body {

    font-family: 'Poppins', sans-serif;

    background: linear-gradient(to bottom, #a8c0ff, #3f2b96);

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

    margin: 0;

    color: white;

    flex-direction: column;

}

.container {

    display: flex;

    max-width: 600px;

    background: rgba(255, 255, 255, 0.1);

    border-radius: 15px;

    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.3);

    overflow: hidden;

    backdrop-filter: blur(10px);

    flex-direction: column;

    align-items: center;

    text-align: center;

    padding: 20px;

}

.weather-side {

    background: linear-gradient(135deg, #6a11cb, #2575fc);

    padding: 30px;

    border-radius: 15px;

}

.weather-city {

    font-size: 40px;

    font-weight: bold;

    text-transform: uppercase;

    margin-bottom: 10px;

    display: flex;

    align-items: center;

    justify-content: center;

    gap: 10px;

}

.weather-temp {

    font-size: 60px;

    font-weight: bold;

}

.weather-desc {

    font-size: 30px;

    font-weight: bold;

    text-transform: capitalize;

}

.weather-wind {

    font-size: 25px;

    font-weight: bold;

    margin-top: 10px;

}

.location-container {

    text-align: center;

    margin-top: 20px;

}

@media (max-width: 768px) {

    .container {

        width: 90%;

    }

}

**Javascript:-**

function login() {

    let username = document.getElementById('username').value;

    let password = document.getElementById('password').value;

    if (username === "admin" && password === "1234") {

        document.getElementById('login-container').style.display = 'none';

        document.getElementById('weather-app').style.display = 'block';

    } else {

        alert("Invalid username or password!");

    }

}

async function updateWeather() {

    let location = document.getElementById("location-input").value;

    let apiKey = "4bfafed8a53dd6ac18047d2095a85fd7";  // Replace with your actual API key - 51cbbf3b388c2b7c667ccef0c7c35be9

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

    try {

        let response = await fetch(url);

        let data = await response.json();

        if (data.cod !== 200) {

            alert(`Error: ${data.message}`);

            return;

        }

        // Get country flag

        let countryCode = data.sys.country.toLowerCase();

        let countryFlag = `https://flagcdn.com/w40/${countryCode}.png`;

        // Update UI

        document.querySelector(".weather-city").innerHTML = `${data.name}, ${data.sys.country} <img src="${countryFlag}" width="40" height="30">`;

        document.querySelector(".weather-temp").textContent = `${data.main.temp}°C`;

        document.querySelector(".weather-desc").textContent = data.weather[0].description;

        document.querySelector(".weather-wind").textContent = `Wind: ${data.wind.speed} km/h`;

    } catch (error) {

        alert("Network error! Please check your internet connection.");

    }

}

function updateDate() {

    let today = new Date();

    let options = { weekday: "long", month: "long", day: "numeric" };

    document.querySelector(".date-dayname").textContent = today.toLocaleDateString("en-US", options);

}

**Results:**

**Key Features and Achievements:**

**Real-time Weather Updates:** Provides instant weather details for any location worldwide.

**Secure Access:** Ensures restricted access through login authentication.

**User-Friendly Interface:** Simple and intuitive UI for seamless interaction.

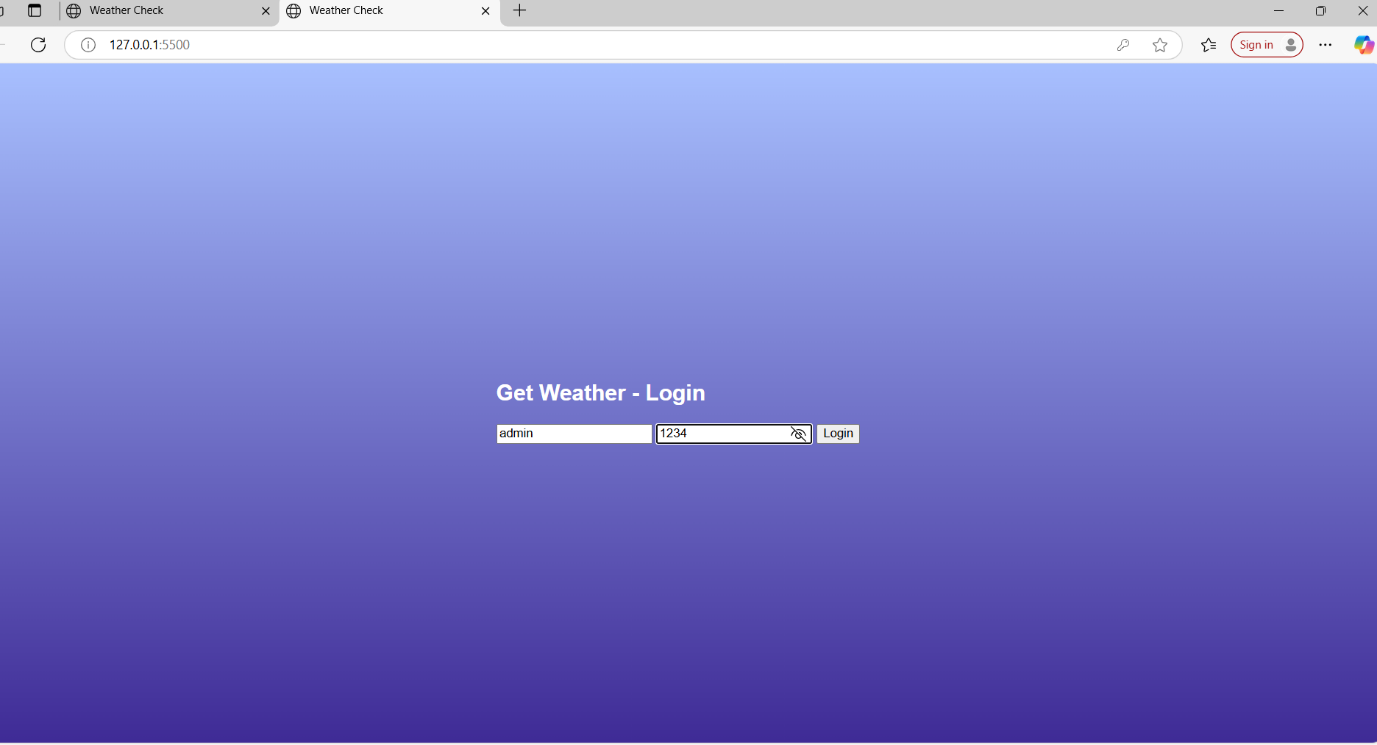
**API Integration:** Uses OpenWeather API for reliable weather data.

**Responsive Design:** Works seamlessly across different devices.

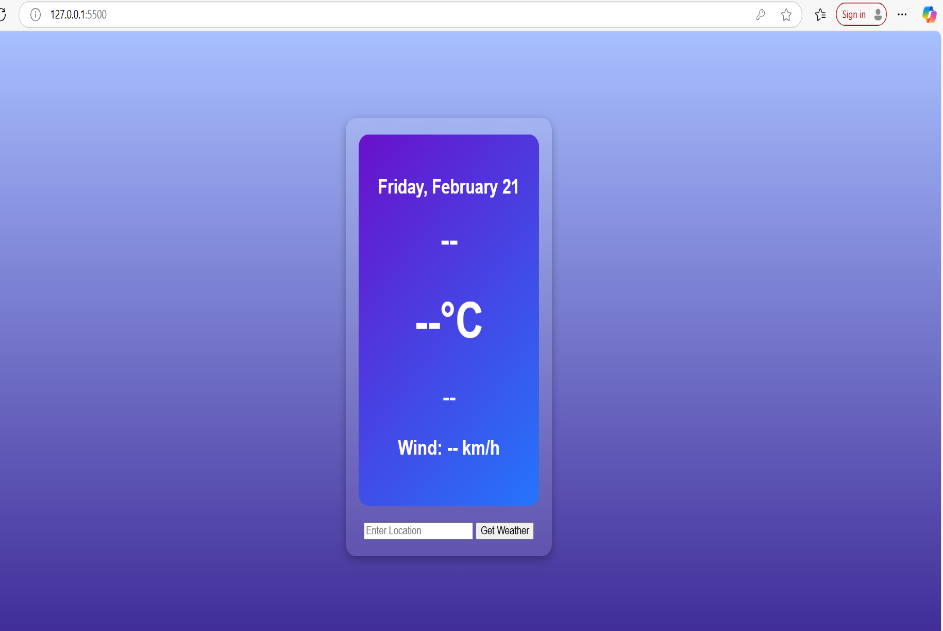
**Dynamic Visuals:** Implements modern UI elements to enhance the user experience.

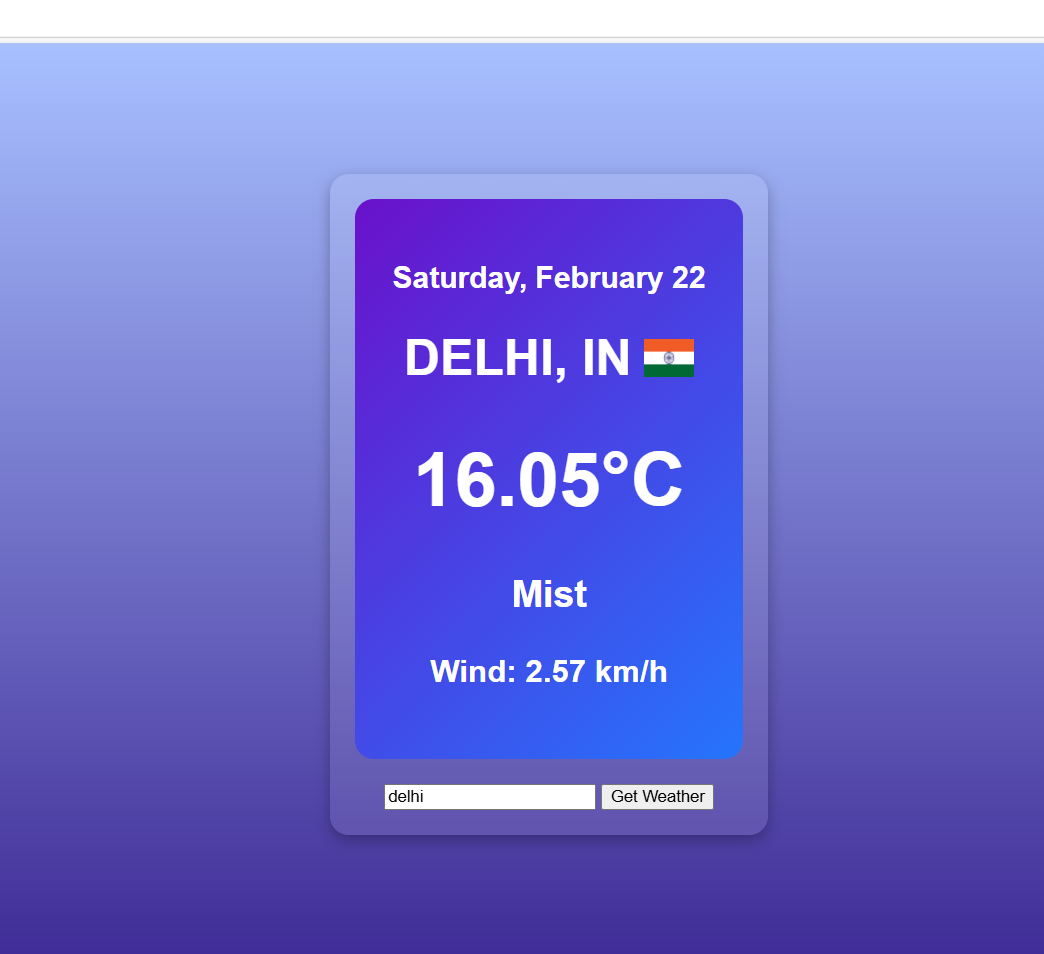
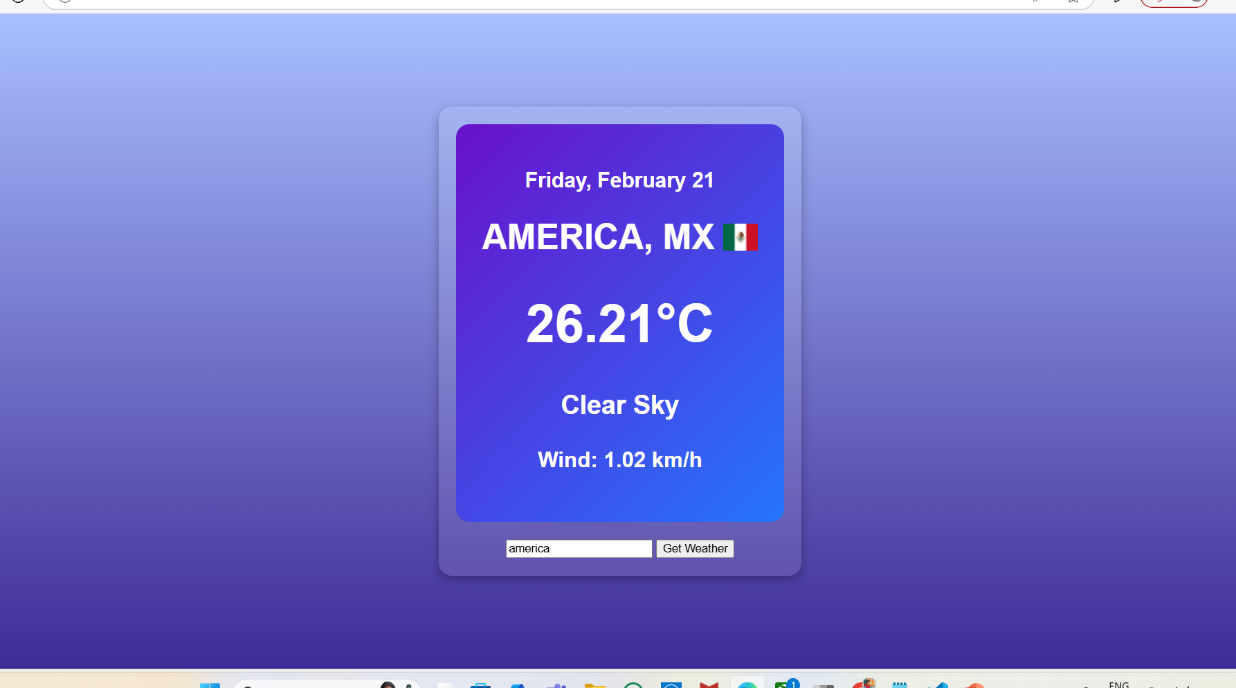
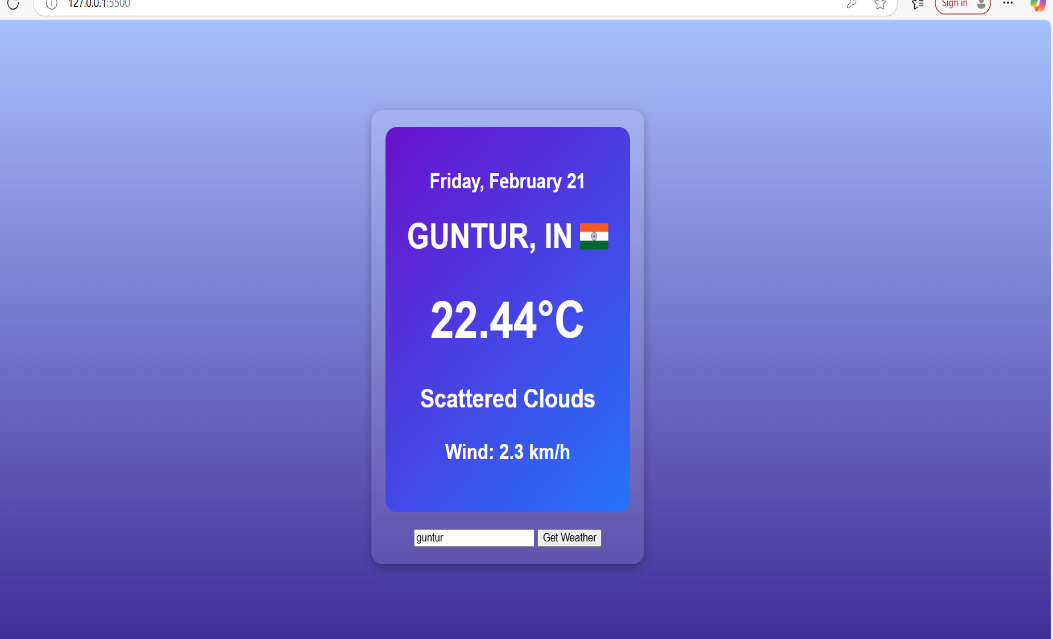
**Output screen(s):**

**Log in page:**

****

**Enter location page:**



**c  **

**Conclusion:-**

The Weather Check Web Application is a simple yet powerful tool that provides real-time weather updates in a user-friendly and visually appealing manner. By integrating the OpenWeather API, the application ensures accurate weather forecasts with minimal user input.

This project successfully addresses the limitations of traditional weather sources by offering a secure login system, a responsive design, and an efficient error-handling mechanism. Users can quickly retrieve weather details such as temperature, weather conditions, wind speed, and country flags in a seamless and interactive way.

Key Achievements:

Real-time weather updates for any location.  
Secure authentication system to restrict unauthorized access.  
 Modern UI/UX design for an intuitive and engaging user experience.  
Cross-platform compatibility, making it accessible on desktops and mobile devices.  
 API-driven data retrieval for accuracy and reliability.  
 Error-handling mechanisms to provide clear and helpful alerts.

.**REFERENCES :**

1. **JavaScript Guide** – Covers JavaScript functions, API requests, and data processing.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

* + 1. **OpenWeather API Documentation** – Provides real-time weather data retrieval and API integration.

<https://openweathermap.org>

1. **JavaScript Guide** – Covers JavaScript functions, API requests, and data processing.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

1. **OpenWeather API Documentation** – Provides real-time weather data retrieval and API integration.

<https://openweathermap.org>

1. **HTML & CSS Tutorials** – Guides for front-end development and responsive UI design.

<https://www.w3schools.com>