Weather Forecasting Application Using Web-Based Model-View and API

MINI PROJECT REPORT

SUBMITTED BY

BARANI KUMAR R
(Reg.No.1913141100009)

Under the guidance of

(Dr.J.Vanathi – Head of the department,

B.Sc. Information Technology)

The partial fulfillment of the requirements
For the award of the degree
Of
Bachelor of Information Technology

DEPARTMENT OF INFORMATION TECHNOLOGY

GURU NANAK COLLEGE (AUTONOMOUS)

Guru Nanak Salai, Velachery, Chennai – 600 042

MAY - 2022

GURU NANAK COLLEGE (Autonomous)

(Affiliated to University of Madras)

Guru Nanak Salai, Velachery, Chennai - 600 042.

This is to certify that, this is a bonafide record of work done by **BARANIKUMAR R** with Register No **1913141100009** in Mini Project during the Academic Year 2021-2022.

Internal Project Guide **Head of the Department**

Submitted for the Fifth Semester Mini Project Examination held on _____ at the BACHELOR OF INFORMATION TECHNOLOGY, GURU NANAK COLLEGE (Autonomous), Guru Nanak Salai, Velachery, Chennai - 600 042.

Internal Examiner **External Examiner**

GURU NANAK COLLEGE (Autonomous)

(Affiliated to University of Madras)

Guru Nanak Salai, Velachery, Chennai - 600 042.

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

I **BARANI KUMAR R, Reg**. No: **19131411000009,** Third year BSc IT student, hereby declare that the Mini Project Report entitled, **Weather Forecasting Application Using Web-Based Model-View and API** is the original work carried out by me under the supervision of **Dr.J.Vanathi – Head of the department (B.Sc. Information Technology),** towards the partial fulfillment of the requirements of BSc IT Degree at **Guru Nanak College (Autonomous).**

I further declare that this Mini Project has not been submitted elsewhere for any other degree.

Place: Chennai Sincerely,

Date:

GURU NANAK COLLEGE (Autonomous)

(Affiliated to University of Madras)

Guru Nanak Salai, Velachery, Chennai - 600 042.

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Mini Project entitled **Weather Forecasting Application Using Web-Based Model-View and APi** Submitted in partial fulfillment of the requirement for the award of Bachelor of Information Technology Degree of Guru Nanak College (Autonomous) is a result of a Bonafide work carried out by **BARANI KUMAR R during** the year 2021-2022, under my supervision.

Place: Chennai **Internal Guide**

Date:

Acknowledgment

I also convey my regards to our **Principal Dr. M.G. Ragunathan f**or their support to pursue this course.

At the outset, I take this opportunity to express my cheerful thanks to our **Head of Department Dr.J.Vanathi M.C.A., M.Phil., Ph.D., and SET** for always being a source of guidance and Inspiration.

I express my deep sense of gratitude to my guide **Dr.J.Vanathi M.C.A., M.Phil., Ph.D., SET.** For his encouragement to this project. They motivated me to do a project using JavaScript, and JSON with Dialog flow API and I was very new to this platform. But he encouraged me all way to complete this project.

ABSTRACT:

Weather is the state of the atmosphere at a given place and time in regards to heat, cloudiness, dryness, sunshine, wind, and rain. Of all the geophysical phenomena weather is the most significant one that influences us. The amount of Rainfall prediction is a major issue for the weather department as it is associated with human life and the economy. Excess rainfall is the major cause of natural disasters such as drought and flood which are encountered by people every year across the world. Weather can vary greatly and largely depends on the climate, seasons, and various other factors. The chief goal of this work is to get the weather forecast of any city throughout the world through an application. we present a novel methodology for improving the performance and dependability of application-level messaging in Grid systems. Based on the Network Weather Service, our system uses nonparametric statistical forecasts of request-response times to automatically determine message timeouts. By choosing a timeout based on predicted network performance, the methodology improves application and Grid service performance as extraneous and overly-long timeouts are avoided. We describe the technique, the additional execution, and programming overhead it introduces, and demonstrate the effectiveness of using a wide-area test application.

TABLE OF CONTENTS

CHAPTER	TOPIC	PAGE NO.
1	**Introduction**	1
	1.1 Objectives	
	1.2 Project Overview	
2	**System Analysis**	
	3.1 Existing System	
	3.2 Proposed System	
3	**System Requirements**	
	4.1 Hardware Configuration	
	4.2 Software Configuration	
4	**System Design**	
	5.1 Input Design	
	5.2 Output Design	
	5.3 Code Design	
	5.4 Data Flow Diagram	
5	**System Testing** **(YOU CAN USE TEST CASES USED IN YOUR PROJECT)**	
	6.1 White Box Testing	
	6.2 Block Box Testing	
	6.3 Unit Testing	
6	**Conclusion**	
7	**Appendices**	
	9.1 Sample Code	
	9.2 Screen Shots (example : login page , transaction page)	
8	**References**	

CHAPTER 1

PROJECT INTRODUCTION

1.1 OBJECTIVES:

- Weather conditions affect the entire economy in many ways both directly and indirectly, to do better weather forecasts bring economic opportunities to almost every sector of the economy.

- Helps people prepare for how to dress (i.e. warm weather, cold weather, windyweather, rainy weather)

- Weather forecasts are critical to the commercial and private transportation sector, including airline, shipping, and trucking industries, nationally and internationally.

- Airlines, for example, rely on short-term forecasts to best position their aircraft and adjust flight routes.

- To help aviation meteorologists issue accurate weather forecasts.

- weather is today it also helps us to plan our trips accordingly, it is used for farmers to grow their crops as the weather predicts to enable the aviation industry to make a decision on flight control based on the information given.

- Helps businesses plan for transportation hazards that can result from the weather(i.e. fog, snow, ice, storms, clouds as it relates to driving and flying for example)

- To study and use advanced programming language as a logical tool for forecasting weather conditions.

- Helps farmers and gardeners plan for crop irrigation and protection (irrigationscheduling, freeze protection)

- To make sure of effective analysis, design implementation, and also provide solutions to state problems in Aviation weather forecast.

1.2 PROJECT OVERVIEW:

Forecasters produce text-based, reasonable weather-element forecast outputs (e.g. maximum/minimum temperature, cloud cover) utilizing numerical weather prediction (NWP) output as guidance in the classical forecast procedure used by most NMHSs. Schedule-driven, product-oriented, and labor-intensive processes are typical. Technological advancements and scientific breakthroughs have allowed NMHS' hydrometeorological forecasts and warnings to become significantly more precise during the previous decade. Customers and partners of the National Weather Service (NWS) demanded precise predictions in gridded, digital, and graphic formats as computer technology and high-speed dissemination channels (e.g., the Internet) advanced. The amount of additional information that can be provided to the user community is limited to traditional NWS text forecast products. The concept of digital database forecasting gives you the flexibility to address customer/partner requests for more precise, detailed hydrometeorological forecasts. One of the most exciting opportunities to connect with PWS is through digital database forecasting.

CHAPTER 2

SYSTEM ANALYSIS

2.1 Existing System:

As computer technology and high-speed dissemination systems evolved (e.g.Internet), National Weather Service (NWS) customers/partners were demanding detailed forecasts in gridded, digital, and graphic formats. Traditional NWS text forecast products limit the amount of additional information that can be conveyed to the user community. The concept of digital database forecasting provides the capability to meet customer/partner demands for more accurate, detailed hydrometeorological forecasts. Digital database forecasting also offers one of the most exciting opportunities to integrate PWS forecast dissemination and service delivery, which most effectively serves the user community.

2.2 Proposed System:

We propose a system that will predict weather from previous data in the database. Fetch the user's device position. If you're unable to get the user's position, set London as the default location. Fetch weather data (forecast data and current weather data) of the location from weather API Format the data into charts Display the chart and current weather.

CHAPTER 3

SYSTEM REQUIREMENTS

The product will be operating in windows with any web browser with newer version. The hardware configuration includes Hard Disk:40 GB, Monitor:15" IPS LCD monitor, and Keyboard. The basic input devices required are a

keyboard, mouse, and output devices are a monitor, mobile devices, etc.

3.1 HARDWARE REQUIREMENTS

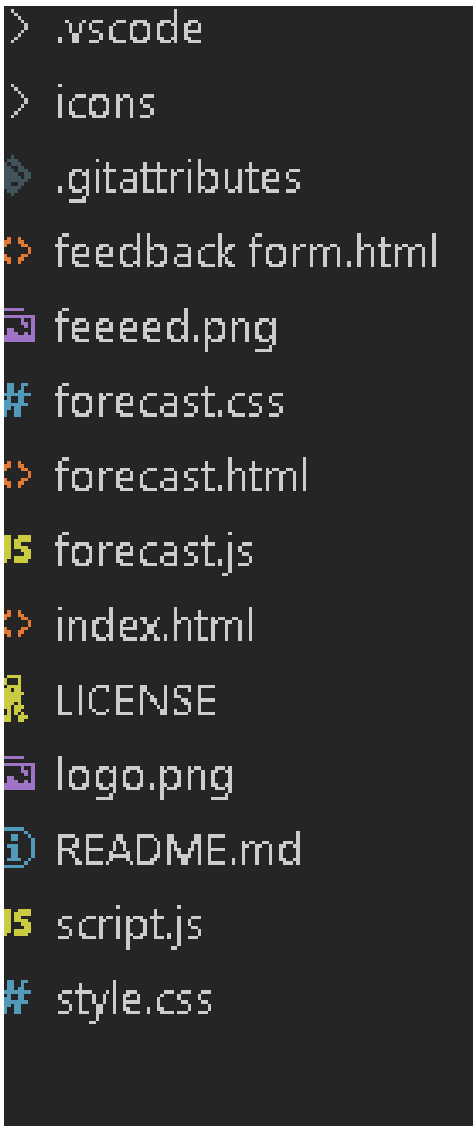
Operating system	:	Windows & latest web browser
RAM	:	3GB (minimum requirement)
Hard Disk	:	12GB working space (minimum requirement)

3.2 SOFTWARE REQUIREMENTS

Languages	:	CSS, Javascript, html,Bootstrap
Tools	:	Visual studio code, Javascript (ES6)
Technology	:	weather api, interface scripts

CHAPTER 4

SYSTEM DESIGN



4.1 INPUT DESIGN:

In this, we use the CSS transition property to animate the width of the search input when it gets focused. You will learn more about the transition property later, in our CSS Transitions chapter.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.[14] It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O. JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

4.2 OUTPUT DESIGN:

JavaScript Output defines the ways to display the output of a given code. The output can be displayed by using four different ways which are listed as inner HTML It is used to access an element. It defines the HTML content using Weather APIs are Application Programming Interfaces that provide access to current & historical weather data on a global scale. There are several public weather APIs like OpenWeatherMap API, Dark Sky API, & Visual Crossing Weather API. On a single search user can get desired

results.

4.3 CODE DESIGN:

To request a user location (lat/long coordinates) within the web browser we will leverage the HTML5 GeoLocation API. This functionality is available in all major browsers (desktop and mobile) and includes IE - yeah, crazy I know. But just because it's supported, doesn't mean our code will always bring back a valid location.

1. Get user coordinates using the HTML5 GeoLocation API:

```
//Check if the geolocation API exists
if (navigator.geolocation) {
  //true
  alert('Lets get the location (placeholder)');
} else {
  //false
  alert('geolocation not available?! What browser is this?');
  // prompt for city?
}
```

Here is a short sample with no error checks with an inline success callback so we can easily identify the basics:

```
//Short sample version with inline success callback
if (navigator.geolocation) {
    //Initial a request for the location
    navigator.geolocation.getCurrentPosition(function(pos){
      //'pos' return object has many properties we can grab
      var geoLat = pos.coords.latitude.toFixed(5);
      var geoLng = pos.coords.longitude.toFixed(5);
      var geoAcc = pos.coords.accuracy.toFixed(1);
    });
}
```

2. Request weather data for the coordinates using an online weather API:

There are many weather APIs available, and most of them have a free tier (for low usage and/or testing). Each will have some pros and cons, including various advanced features (15-day forecast, multiple forecast models, etc). One major change in this area is the recent shutdown of the very popular Yahoo! Weather API, probably one of the most used weather APIs over the past decade. As of January 3rd, 2019 the service is offline with a replacement service just starting its "by-request onboarding" phase. Since that's currently not available, we are going to use DarkSky.

```
<!-- API Params -->
https://api.darksky.net/forecast/[key]/[latitude],[longitude]
<!-- Sample Request -->
https://api.darksky.net/forecast/myFakeKey123abc/43.642567,-79.387054
```

Building the request in JavaScript With the DarkSky URL syntax ready to go, we just need to make the call in our code.

```
_dsSecret = "yourSecret"; //Again, for testing only, should be hidden in proxy

function fn_getWeatherByLL(geoLat,geoLng){
  //API Variables
  var proxy = 'https://cors-anywhere.herokuapp.com/';
  var dsAPI = "https://api.darksky.net/forecast/";
  var dsKey = _dsSecret + "/";
  var dsParams = "?exclude=minutely,hourly,daily,alerts,flags&units=auto";
  //Concatenate API Variables into a URLRequest
  var URLRequest = proxy + dsAPI + dsKey + String(geoLat) + "," + String(geoLng) +
dsParams
```

```
  //Make the jQuery.getJSON request
  $.getJSON( URLRequest )
    //Success promise
    .done(function( data ) {
      var wSummary = data.currently.summary;
      var wTemperature = data.currently.temperature;
      // lots of results available on the data object
      // use the results to populate the GUI here
    })
    //Error promise
    .fail(function() {
      alert('Sorry, something bad happened when retrieving the weather');
    }
  );
}
```

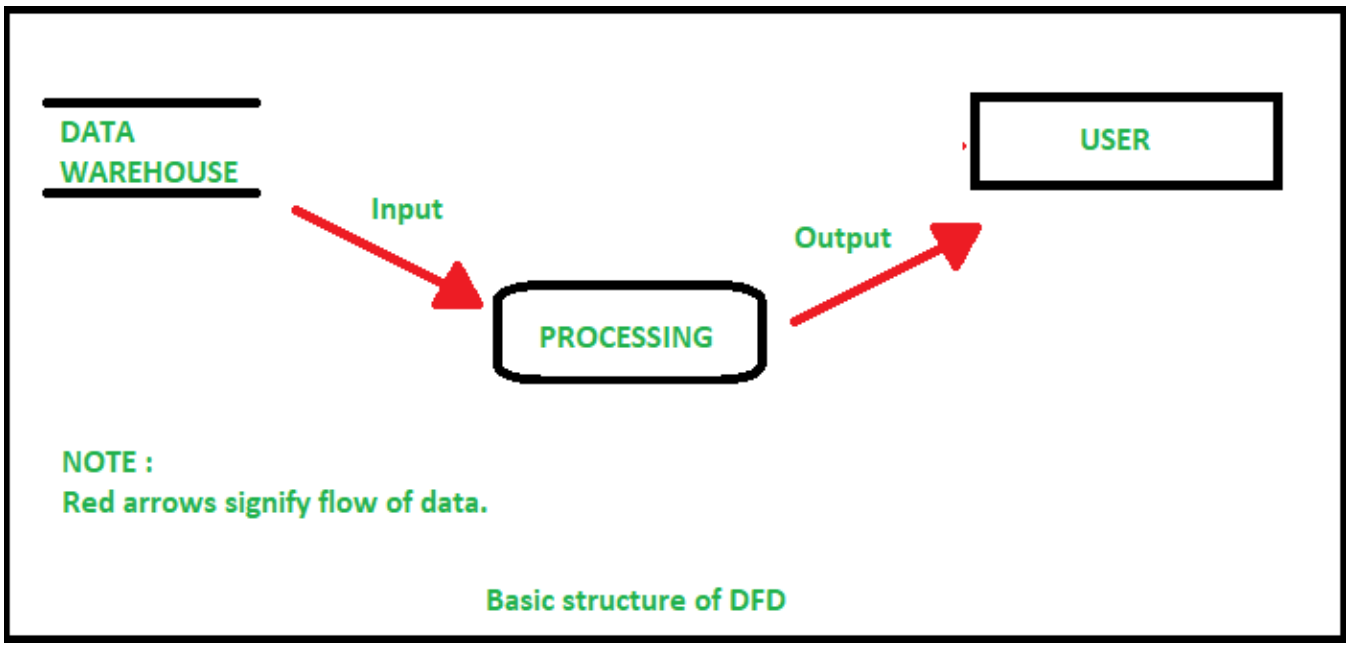
User interface code that data object contains all the information needed to build your weather dashboard. From current temperature, daily max/mins, and short-term forecast

3. Reverse Geocode the coordinates to get additional location details:

For weather data, we don't need amazing accuracy because weather data doesn't change at the street level. So even if the accuracy sucks, it should have little consequence. But if you want to find and show the current city, address, or include a pin on a map, reverse geocoding to the rescue. Using a Reverse Geocoding API is pretty much identical to using a weather API - Provide coordinates in a URL and wait for results.

```
<!-- Sample request -->
https://nominatim.openstreetmap.org/reverse?format=jsonv2&lat=43.642567&lon=-
79.387054
```

DATA FLOW DIAGRAM:



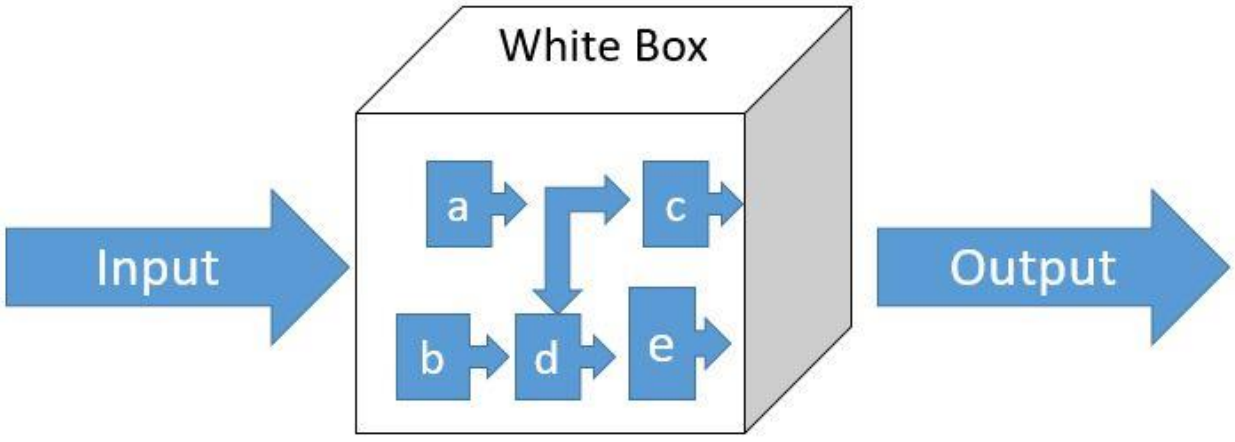
Basic structure of DFD

CHAPTER 5

APPENDICES

5.1 WHITEBOX TESTING:

White Box Testing is a software testing technique in which the internal structure, design, and coding of software are tested to verify flow of input-output and to improve design, usability, and security. In white-box testing, code is visible to testers so it is also called Clear box testing, Open box testing, transparent box testing, Code-based testing and Glass box testing. It is one of two parts of the Box Testing approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, White box testing in software engineering is based on the inner workings of an application and revolves around internal testing. The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings. Likewise, the "black box" in "Black Box Testing" symbolizes not being able to see the inner workings of the software so that only the end-user experience can be tested.



CHAPTER 6

6.0 CONCLUSION:

Weather forecasts are made by collecting quantitative data about the current state of the atmosphere and using a scientific understanding of atmospheric processes to project how the atmosphere will evolve. Weather warnings are important forecasts because they are used to protect life and property. Forecasts based on temperature and precipitation are important to agriculture, and therefore to traders within commodity markets. IT will display current weather including weather conditions, temperature, humidity, wind speed, and date and time. Visualization to display the temperature change, humidity change, and weather conditions of each day.

SAMPLE CODE:

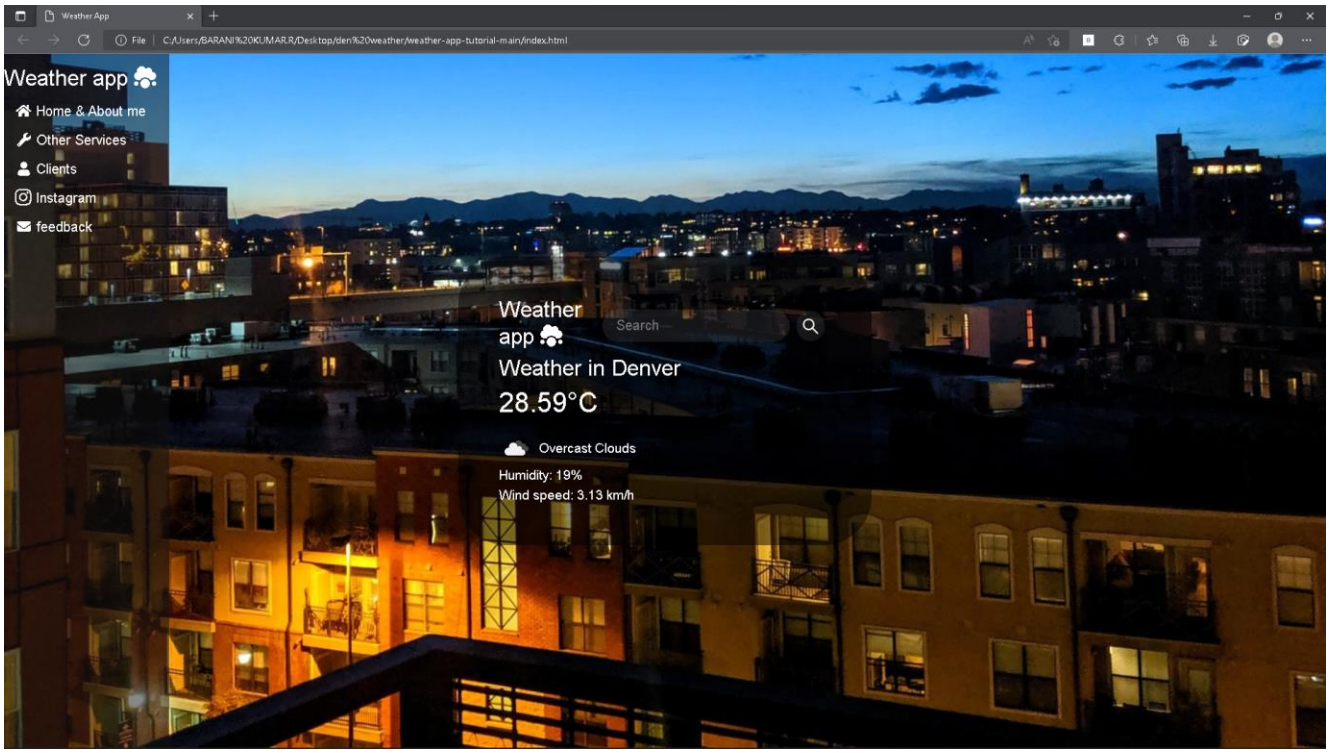
```
let weather = {
  apiKey: "b06d9a690d342de66ecc95ad8c8b574f",
  fetchWeather: function (city) {
    fetch(
      "https://api.openweathermap.org/data/2.5/weather?q=" +
        city +
        "&units=metric&appid=" +
        this.apiKey
    )
      .then((response) => {
        if (!response.ok) {
          alert("No weather found.");
          throw new Error("No weather found.");
        }
        return response.json();
      })
      .then((data) => this.displayWeather(data));
```

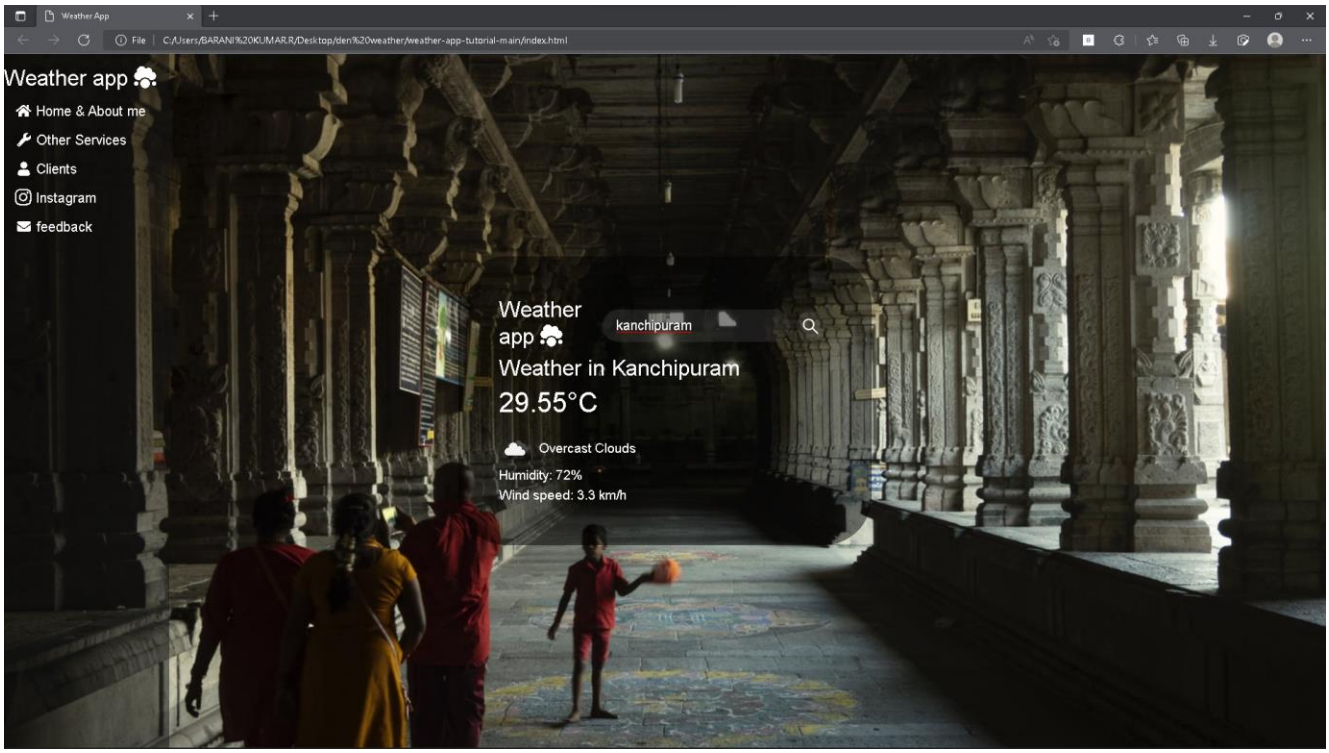
```
  },
  displayWeather: function (data) {
    const { name } = data;
    const { icon, description } = data.weather[0];
    const { temp, humidity } = data.main;
    const { speed } = data.wind;
    document.querySelector(".city").innerText = "Weather in " + name;
    document.querySelector(".icon").src =
      "https://openweathermap.org/img/wn/" + icon + ".png";
    document.querySelector(".description").innerText = description;
    document.querySelector(".temp").innerText = temp + "°C";
    document.querySelector(".humidity").innerText =
      "Humidity: " + humidity + "%";
    document.querySelector(".wind").innerText =
      "Wind speed: " + speed + " km/h";
    document.querySelector(".weather").classList.remove("loading");
    document.body.style.backgroundImage =
      "url('https://source.unsplash.com/1600x900/?" + name + "')";
  },
  search: function () {
    this.fetchWeather(document.querySelector(".search-bar").value);
  },
};

document.querySelector(".search button").addEventListener("click", function () {
  weather.search();
});

document
  .querySelector(".search-bar")
  .addEventListener("keyup", function (event) {
    if (event.key == "Enter") {
      weather.search();
    }
  });
```

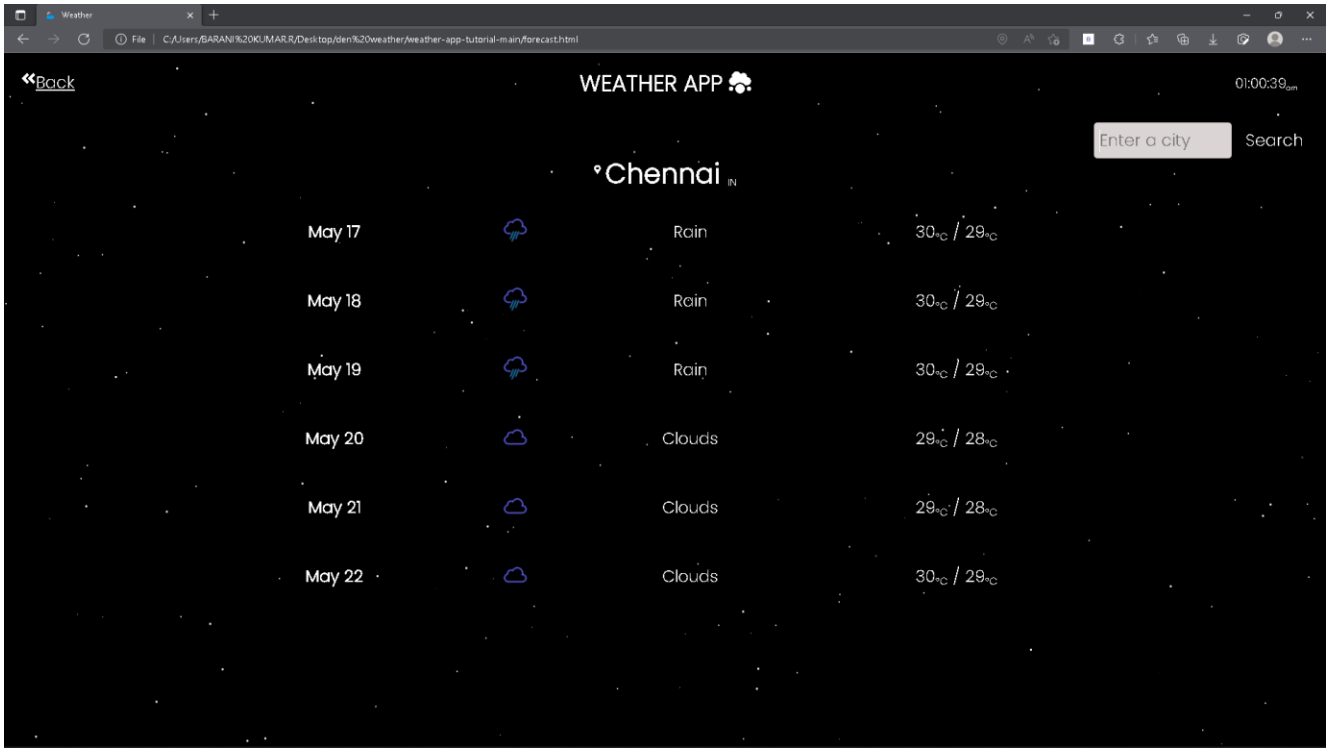
```
weather.fetchWeather("Denver");

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));
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.substr(
    "cdn.weatherapi.com/weather/64x64/".length);
    icon.src = "./icons/" + iconId;
    cloudoutput.innerHTML = data.current.cloud + "%";
    humidityOutput.innerHTML = data.current.humidity + "%";
    windOutput.innerHTML =data.current.wind_kph + "km/h";

    let timeOfDay = "day";
    const code = data.current.condition.code;

    if(data.current.is_day) {
        timeOfDay + "night";

    }
    if(code ==1000) {
        app.style.background =`url(./images/($timeOFday}/clear.jpg)`;
        btn.style.background = "#e5ba92";
        if(timeOFday == "night") {
            btn.style.background = "#181e27";
        }
    }
    else if (
```

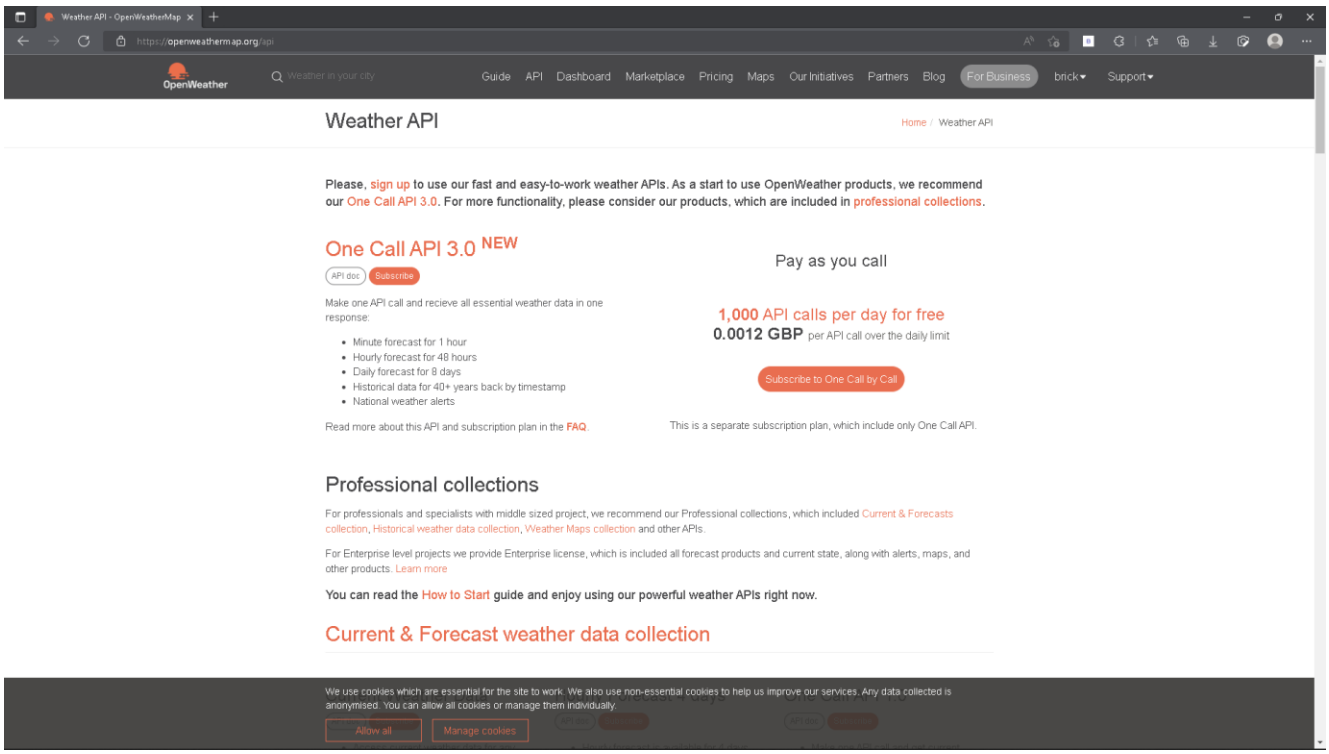
```
        code == 1003 ||
        code == 1006 ||
        code == 1009 ||
        code == 1030 ||
        code == 1069 ||
        code == 1087 ||
        code == 1035 ||
        code == 1073 ||
        code == 1076 ||
        code == 1079 ||
        code == 1082

    ) {
        app.style.backgroundImage = `url(./images/${timeOFday}/cloudy.jpg)`;
        btn.style.background = "#fa6d1b";
        if(timeOfday == "night")
        {
            btn.style.background = "#181e27";
        }
     }
```

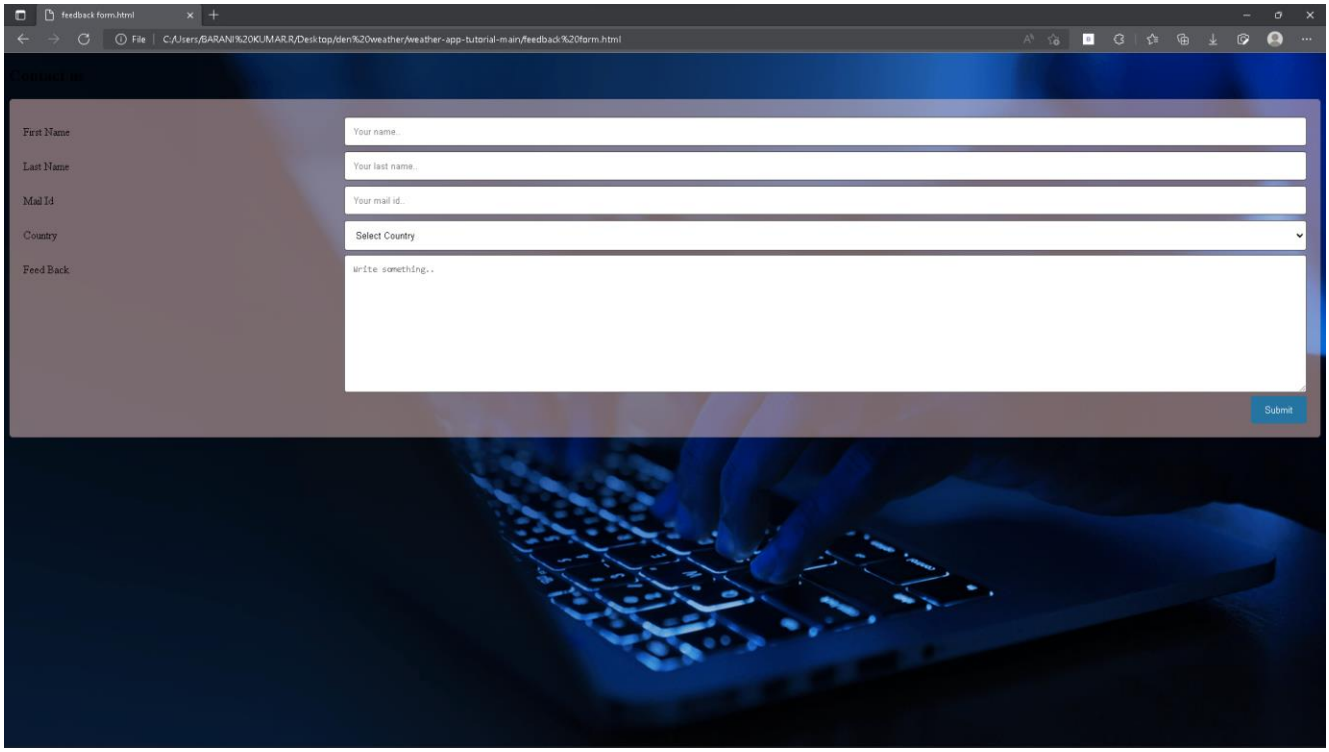
SCREENSHOT:


Weather App
C:/Users/BARANI%20KUMAR.R/Desktop/den%20weather/weather-app-tutorial-main/index.html
Weather app
Home & About me
Other Services
Clients
Instagram
feedback
Weather app
Search
Weather in Denver
28.59°C
Overcast Clouds
Humidity: 19%
Wind speed: 3.13 km/h


Weather App
C:/Users/BARANI%20KUMAR.R/Desktop/den%20weather/weather-app-tutorial-main/index.html
Weather app
Home & About me
Other Services
Clients
Instagram
feedback
Weather app
kanchipuram
Weather in Kanchipuram
29.55°C
Overcast Clouds
Humidity: 72%
Wind speed: 3.3 km/h


Weather
File | C:/Users/BARANI%20KUMAR.R/Desktop/den%20weather/weather-app-tutorial-main/forecast.html
Back
WEATHER APP
01:00:39
Enter a city
Search
Chennai IN
May 17 Rain 30°C / 29°C
May 18 Rain 30°C / 29°C
May 19 Rain 30°C / 29°C
May 20 Clouds 29°C / 28°C
May 21 Clouds 29°C / 28°C
May 22 Clouds 30°C / 29°C


Weather API - OpenWeatherMap
https://openweathermap.org/api
OpenWeather
Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For Business brick Support
Weather API
Home / Weather API
Please, sign up to use our fast and easy-to-work weather APIs. As a start to use OpenWeather products, we recommend our One Call API 3.0. For more functionality, please consider our products, which are included in professional collections.
One Call API 3.0 NEW
API doc Subscribe
Make one API call and recieve all essential weather data in one response:
Minute forecast for 1 hour
Hourly forecast for 48 hours
Daily forecast for 8 days
Historical data for 40+ years back by timestamp
National weather alerts
Read more about this API and subscription plan in the FAQ.
Pay as you call
1,000 API calls per day for free
0.0012 GBP per API call over the daily limit
Subscribe to One Call by Call
This is a separate subscription plan, which include only One Call API.
Professional collections
For professionals and specialists with middle sized project, we recommend our Professional collections, which included Current & Forecasts collection, Historical weather data collection, Weather Maps collection and other APIs.
For Enterprise level projects we provide Enterprise license, which is included all forecast products and current state, along with alerts, maps, and other products. Learn more
You can read the How to Start guide and enjoy using our powerful weather APIs right now.
Current & Forecast weather data collection
We use cookies which are essential for the site to work. We also use non-essential cookies to help us improve our services. Any data collected is anonymised. You can allow all cookies or manage them individually.
Allow all
Manage cookies


feedback form.html
File | C:/Users/BARANI%20KUMAR.R/Desktop/den%20weather/weather-app-tutorial-main/feedback%20form.html
First Name
Your name..
Last Name
Your last name..
Mail Id
Your mail id..
Country
Select Country
Feed Back
Write something..
Submit

8.0 References:

[1]. Venkatesh, R., C. Balasubramanian, and M. Kaliappan. "Rainfall prediction using generative adversarial networks with convolution neural network." Soft Computing: 1-14.

[2]. Naidu, D., Majhi, B., & Chandniha, S. K. (2021). Development of Rainfall Prediction Models Using Machine Learning Approaches for Different Agro-Climatic Zones. In Handbook of Research on Automated Feature Engineering and Advanced Applications in Data Science (pp. 72-94). IGI Global.

[3]. Coban, Veysel, Ezgi Guler, Taner Kilic, and Suheyla Yerel Kandemir. "Precipitation forecasting in Marmara region of Turkey." Arabian Journal of Geosciences 14, no. 2 (2021): 1-10.

[4]. Kumar, Vikram, Manvendra Singh Chauhan, and Shanu Khan. "Application of Machine Learning Techniques for Clustering of Rainfall Time Series Over Ganges River Basin."