WEATHER FORCASTING WEB APP

A Minor Project-II (AD608)

Submitted in partial fulfillment of the requirements for the
Award of Degree of
Bachelor of Technology in Artificial intelligence and data science

Submitted to

RAJIV GANDHI PROUDYOGIKI VISHWAVIDHYALAYA,
BHOPAL (M.P.)

Submitted by

ISHA BISEN (0111AS211022)
BHAVYA DATTEY (0111AS211015)
PRADEEP PAWAR (0111AS211038)
HIMANSHU SAHU (0111AS211021)

Under the Guidance of

PROF. SAGAR CHOUDHARY

(Associate Professor, CSE Department)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
TECHNOCRATS INSTITUTE OF TECHNOLOGY, BHOPAL
Session: Jan-June 2024

TECHNOCRATS INSTITUTE OF TECHNOLOGY, BHOPAL

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

*This is to certify that the work embodies in this Minor Project-II (CS608) work entitled "*WEATHER FORCASTING WEB APP*" being submitted by ISHA BISEN(0111AS211021) , BHAVYA DATTEY (0111AS211015), PRADEEP PAWAR(0111AS211038),HIMANSHU SAHU(0111AS211021) , in partial fulfillment of the requirement for the award of Bachelor of Technology in Artificial Intelligence and Data Science to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)during the academic session Jan-June 2024 is a record of bonafide piece of work, carried out by him under my supervision and guidance in the Artificial Intelligence and Data Science, Technocrats Institute of Technology, Bhopal.*

Guided by:	**Approved by:**
PROF. SAGAR CHOUDHARY	**Dr. VIVEK SHARMA**
Associate Professor	**Head of the Department**
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE	**ARTIFICAL INTELLIGENCE AND DATA SCIENCE**

TECHNOCRATS INSTITUTE OF TECHNOLOGY, BHOPAL

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

DECLARATION

We, ISHA BISEN (0111AS211022) , BHAVYA DATTEY (0111AS211015), PRADEEP PAWAR(0111AS211038) , HIMANSHU SAHU (0111AS211021), students of Bachelor of Technology in Artificial Intelligence and Data Science, Academic session Jan-June 2024 Technocrats Institute of Technology, Bhopal M.P., hereby declare that the work presented in this Minor Project-II (AD608) Report entitled “WEATHER FORCASTING WEB APP” is the outcome of our own work, is bona fide and correct to the best of our knowledge and this work has been carried out taking care of Engineering Ethics.

ISHA BISEN (0111AS211022)
BHAVYA DATTEY (0111AS211015)
PRADEEP PAWAR (0111AS211038)
HIMANSHU SAHU (0111AS211021)

ACKNOWLEDGEMENT

I deem it’s my privilege to extent my profound gratitude and appreciation towards all those who have directly or indirectly involved themselves in making this project a great success. It gives me immense pleasure to express my deepest sense of gratitude and sincere thanks to my respected guide Dr. Ravindra Yadav for his valuable guidance, encouragement and help for this work.

I would also like to thank Dr. Vivek Sharma, Head of Department Artificial Intelligence & Data Science for providing me with all the moral support and necessary help.

I would also like to express my sincere thanks to Dr. Shashi Kumar Jain, Director Technocrats Institute of Technology, Bhopal for his encouragement and support.

I am also thankful to all the staff members of the Institute for their cooperation in my work.
My sincere appreciation and thanks to my family members and friends for keen interest, continued encouragement and support.

CONTENT

OBJECTIVE

I. **Provide Accurate Weather Forecasts**: Deliver reliable and up-to-date weather information to help users plan their daily activities and long-term events.

II. **Real-Time Weather Updates**: Offer real-time weather updates to keep users informed about sudden changes in weather conditions.

III. **Severe Weather Alerts**: Notify users of severe weather conditions, such as storms, hurricanes, or extreme temperatures, to ensure their safety.

IV. **Localized Weather Information**: Offer weather forecasts tailored to specific locations, allowing users to get relevant information for their area.

V. **User-Friendly Interface**: Create an intuitive and easy-to-navigate website design to enhance user experience and accessibility.

VI. **Educational Content**: Provide articles, videos, and infographics that educate users about weather phenomena, climate change, and related topics.

VII. **Interactive Features**: Include interactive maps, radar images, and weather widgets to help users visualize weather patterns and trends.

VIII. **Historical Weather Data**: Archive past weather data for users interested in historical weather trends and analysis.

IX. **Integration with Other Services**: Allow integration with other applications and services, such as calendar apps or travel planning tools, for a more comprehensive user experience.

X. **Community Engagement**: Foster a community of weather enthusiasts by enabling user-generated content, such as weather reports, photos, and discussion forums.

SCOPE

1. **Geographical Coverage**: Provide weather information for multiple regions, from local neighborhoods to global cities, ensuring comprehensive geographical coverage.

2. **Forecast Duration**: Offer weather forecasts for various timeframes, including hourly, daily, weekly, and extended forecasts up to 15 days or more.

3. **Weather Parameters**: Cover a wide range of weather parameters, such as temperature, humidity, wind speed, precipitation, UV index, and air quality.

4. **Real-Time Data Integration**: Integrate with real-time data sources, such as meteorological agencies and weather stations, to provide up-to-the-minute weather information.

5. **Visualization Tools**: Include tools like interactive maps, satellite imagery, and radar animations to help users visualize weather conditions and trends.

6. **Custom Alerts and Notifications**: Enable users to set up custom alerts for specific weather conditions, such as severe weather warnings, temperature thresholds, or precipitation alerts.

7. **Historical Data Access**: Provide access to historical weather data for users interested in analyzing past weather patterns and trends.

8. **Mobile and Desktop Compatibility**: Ensure the website is fully responsive and accessible on various devices, including smartphones, tablets, and desktop computers.

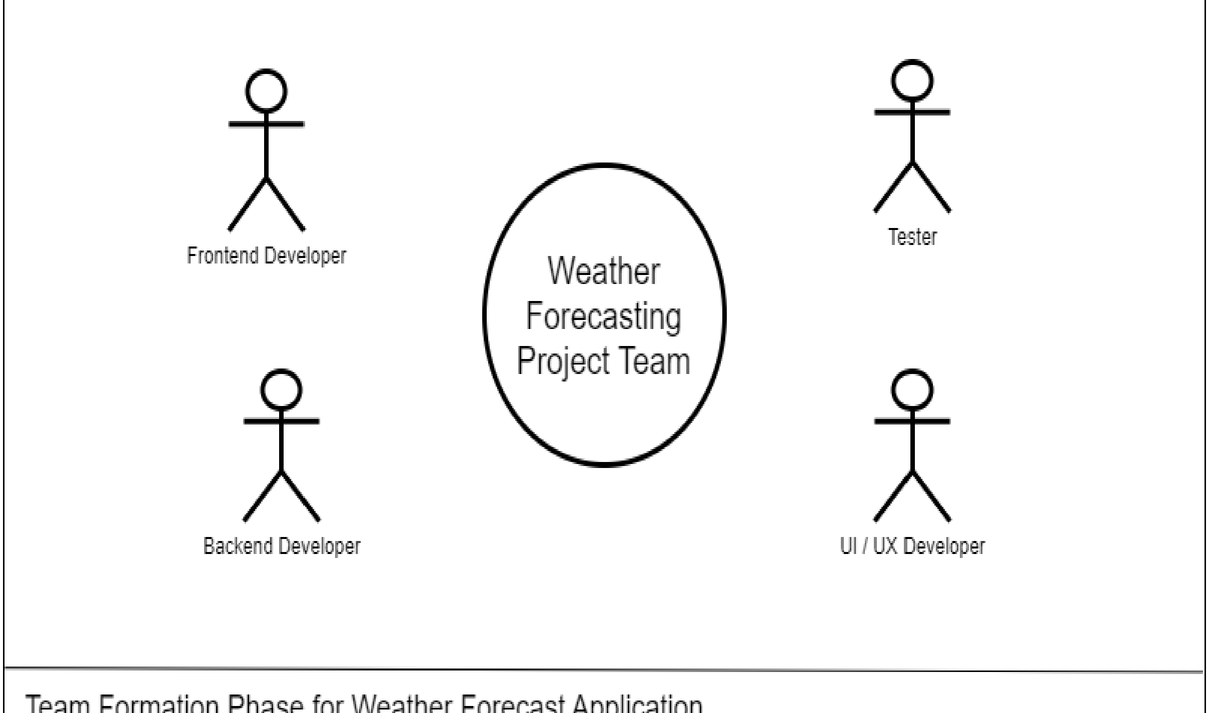
9. **User Personalization**: Allow users to personalize their experience by saving favorite locations, setting preferences for units of measurement, and customizing the display of weather information.

10. **Additional Content and Features**: Offer related content such as weather news, educational articles, climate change information, travel weather guides, and community forums for user engagement and information sharing.

Team Formation Phase

Team formation is a crucial step in any project it significantly impact on your project . In our project as we will be exploring about the web application for weather app so will will be going to require following skill sets.

I. Front end Development (Html , CSS).
II. Back end Development (JavaScript).
III. Tester
IV. UI/UX Developer



Team Formation Phase for Weather Forecast Application

PURPOSE

1. **Informing the Public**: Provide accurate and timely weather information to help users make informed decisions about their daily activities and plans.

2. **Enhancing Safety**: Deliver severe weather alerts and warnings to protect users from potential hazards like storms, floods, and extreme temperatures.

3. **Supporting Travel and Commuting**: Offer weather forecasts and real-time updates to assist users in planning their travel routes and commutes safely.

4. **Promoting Preparedness**: Help users prepare for adverse weather conditions by providing advance notice and tips on how to stay safe.

5. **Facilitating Outdoor Activities**: Provide detailed weather forecasts to help users plan and enjoy outdoor activities such as sports, hiking, and events.

6. **Educational Resource**: Serve as an educational tool to teach users about weather patterns, climate change, and meteorological phenomena.

7. **Enhancing User Experience**: Offer a user-friendly platform with customizable features that cater to individual preferences and needs.

8. **Providing Historical Context**: Archive historical weather data to allow users to analyze past weather trends and patterns.

9. **Encouraging Community Engagement**: Foster a community of weather enthusiasts by enabling user-generated content, discussions, and shared weather reports.

10. **Integrating with Other Services**: Allow seamless integration with other applications and services, such as travel planners, event organizers, and personal calendars, to enhance overall user utility.

PROBLEM STATEMENT

These points identify key problems that the weather app website aims to address, ensuring it meets the diverse needs of its users through accurate, timely, and comprehensive weather information and services.

1. **Lack of Accurate Information**: Users often struggle to find reliable and up-to-date weather information, which can lead to poorly informed decisions and safety risks.

2. **Insufficient Real-Time Updates**: Many existing weather platforms fail to provide real-time updates, resulting in outdated information during critical weather changes.

3. **Inadequate Severe Weather Alerts**: Users lack timely and accurate alerts for severe weather conditions, putting them at risk during extreme events like storms or heatwaves.

4. **Poor User Experience**: Complex and unintuitive interfaces of some weather websites can frustrate users, making it difficult for them to find the information they need quickly.

5. **Limited Geographic Coverage**: Some weather services do not offer comprehensive coverage for all locations, especially smaller or remote areas, leaving users without localized information.

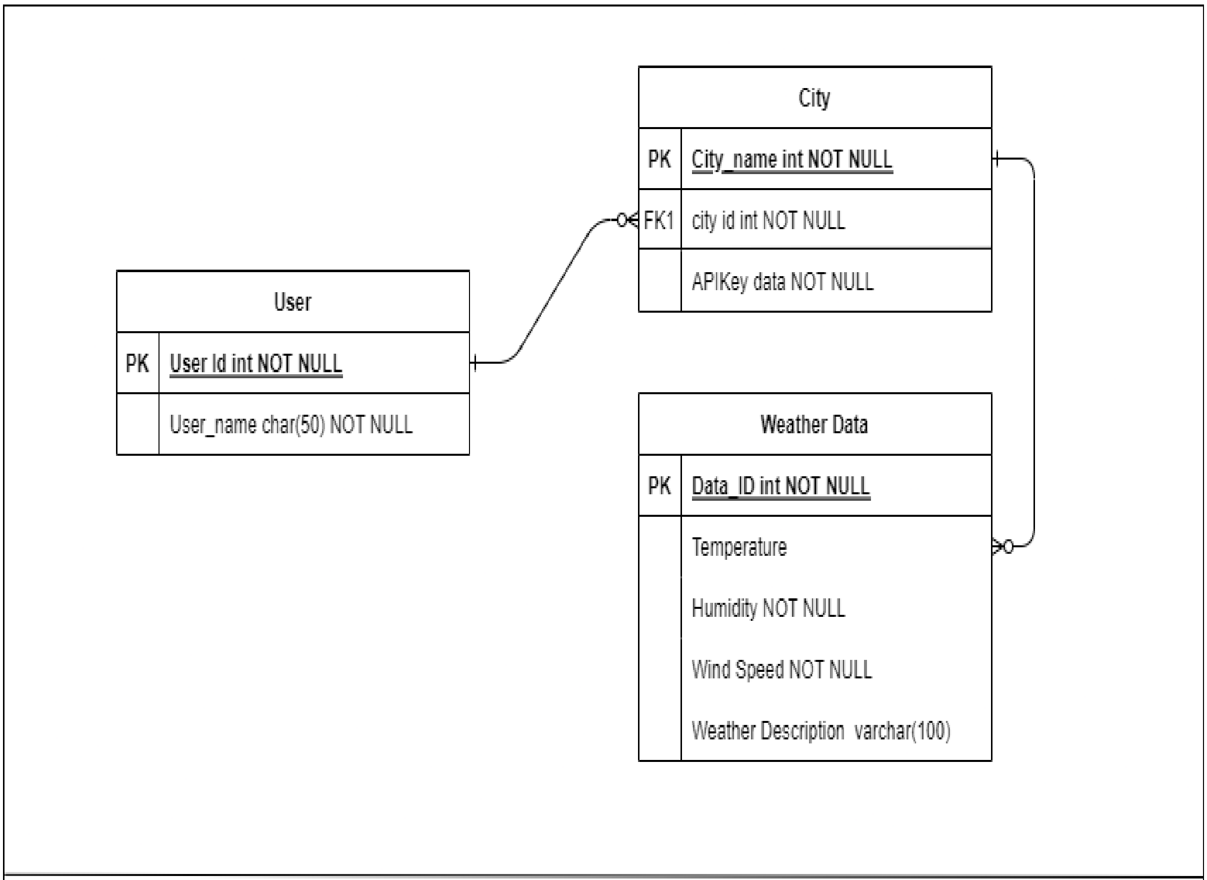
6. **Absence of Customization Options**: Users cannot personalize their weather information according to their preferences, such as setting favorite locations or customizing alerts.

7. **Lack of Detailed Weather Parameters**: Existing services might not provide a full range of weather data, such as humidity, UV index, or air quality, which are essential for specific user needs.

8. **Incompatibility Across Devices**: Many weather websites are not optimized for mobile devices, limiting access for users who rely on smartphones and tablets.

9. **Scarcity of Historical Data**: Users often have no access to historical weather data, which is crucial for research, trend analysis, and planning.

E-R DIAGRAM



ER Diagram for Weather App

Entities:

User: ***Attributes*** User Id (Primary Key)
City***: Attributes*** : City Name (Primary Key) , API Key value.
Weather Details: ***Attributes***: Temperature , Wind Speed , Weather Description , Humidity.

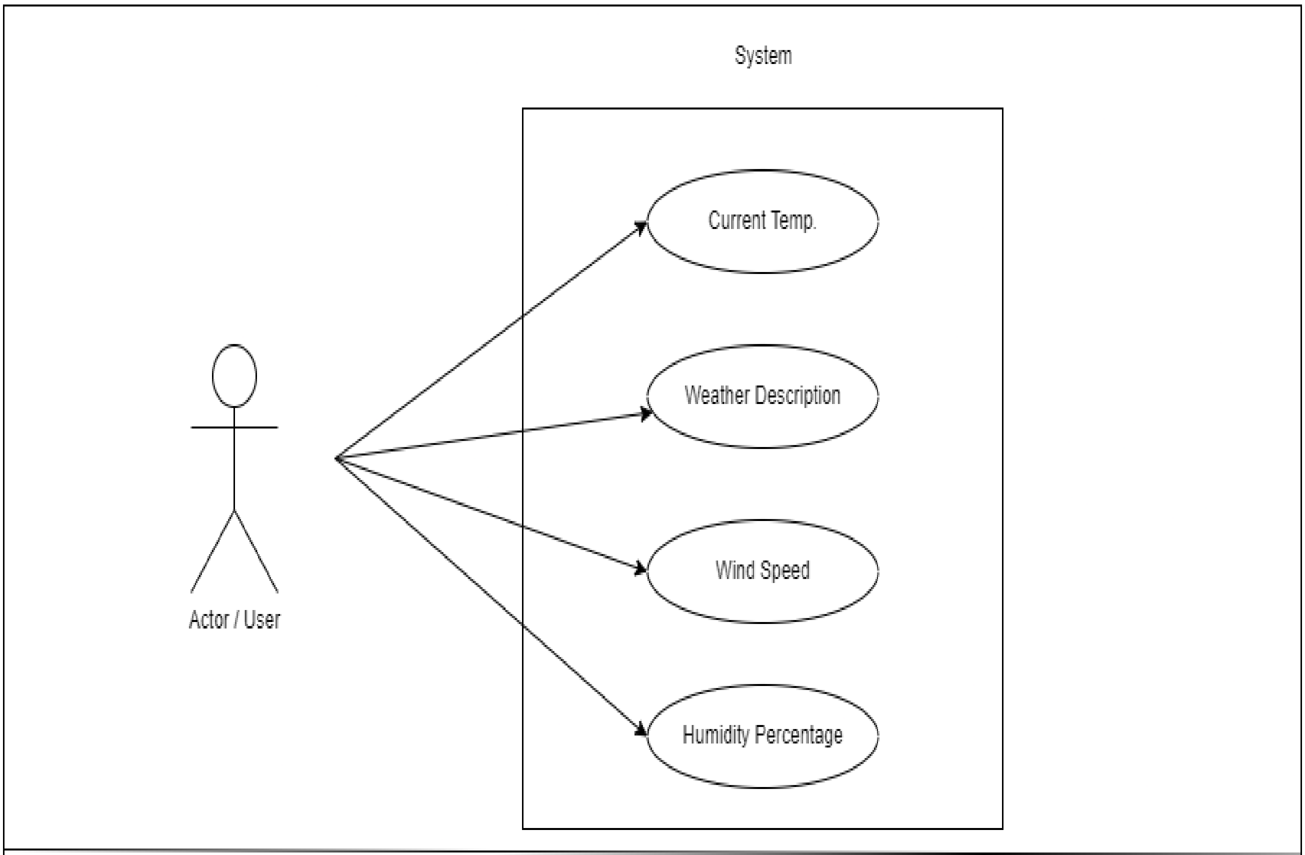
Relation:

Enters: User enters the city name in the application.
Returns: API returns an list of weather details having temperature , wind speed , humidity and weather details

USE CASE DIAGRAM

A use case diagram is a visual representation of the functional requirements of a system, illustrating how users interact with the system and the system's responses. For a weather forecasting app, the use case diagram will include actors such as "User," "Weather Data Provider," or "Administrator".

Here's a simplified use case diagram for a weather forecasting app:



UseCase Diagram for Weather App

Description of the Use Case Diagram:

View Current Weather:
Actor: User
Description: Allows the user to view the current weather conditions for their selected location.

View Hourly Weather Description:
Actor: User
Description: Enables the user to check the weather description for the selected location.

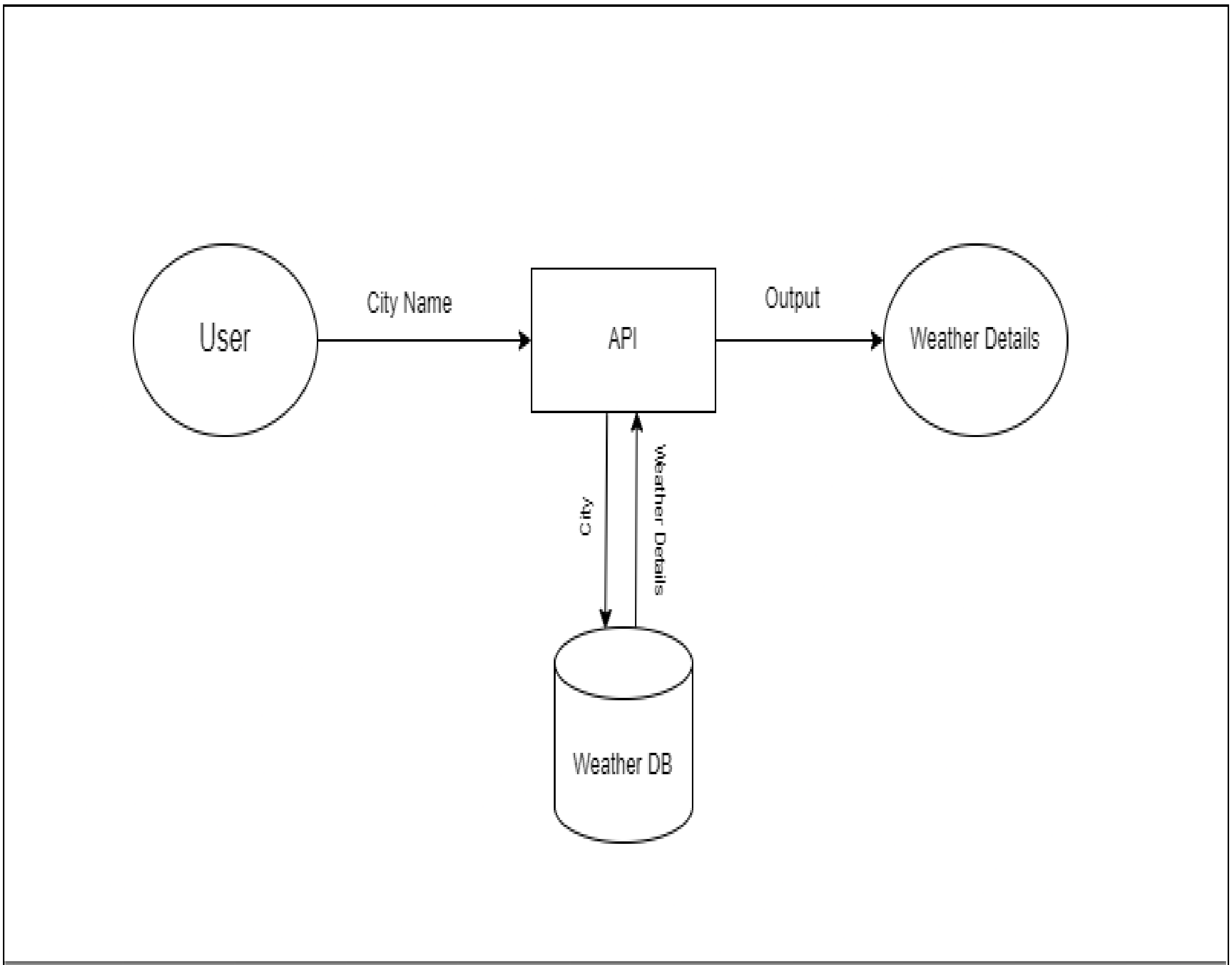
View Daily Forecast:
Actor: User
Description: Permits the user to access the daily weather forecast for the chosen location.

Set Location Preferences:
Actor: User
Description: Allows the user to set and manage location preferences for weather forecasts.

Receive Weather Alerts:
Actor: User
Description: Enables the user to receive alerts for severe weather conditions or customized weather events.

DATA FLOW DIAGRAM

DFD is the abbreviation for Data Flow Diagram. The flow of data in a system or process is represented by a Data Flow Diagram (DFD). It also gives insight into the inputs and outputs of each entity and the process itself. Data Flow Diagram (DFD) does not have a control flow and no loops or decision rules are present. Specific operations, depending on the type of data, can be explained by a flowchart. It is a graphical tool, useful for communicating with users, managers and other personnel. it is useful for analyzing existing as well as proposed systems.



Data Flow Diagram of Weather App

Characteristics of Data Flow Diagram (DFD)

Below are some characteristics of Data Flow Diagram (DFD):

Graphical Representation: Data Flow Diagram (DFD) use different symbols and notation to represent data flow within system. That simplify the complex model.

Problem Analysis: Data Flow Diagram (DFDs) are very useful in understanding a system and can be effectively used during analysis. Data Flow Diagram (DFDs) are quite general and are not limited to problem analysis for software requirements specification.

Abstraction: Data Flow Diagram (DFD) provides a abstraction to complex model i.e. DFD hides unnecessary implementation details and show only the flow of data and processes within information system.

Hierarchy: Data Flow Diagram (DFD) provides a hierarchy of a system. High-level diagram i.e. 0-level diagram provides an overview of entire system while lower-level diagram like 1-level DFD and beyond provides a detailed data flow of individual process.

Data Flow: The primary objective of Data Flow Diagram (DFD) is to visualize the data flow between external entity, processes and data store. Data Flow is represented by an arrow Symbol.

Ease of Understanding: Data Flow Diagram (DFD) can be easily understand by both technical and non-technical stakeholders.

Modularity: Modularity can be achieved using Data Flow Diagram (DFD) as it breaks the complex system into smaller module or processes. This provides easily analysis and design of a system.

TECHNOLOGY USED

HTML (HyperText Markup Language)

Overview:
HTML is the standard markup language used to create web pages. It structures the content on the web and forms the backbone of any web application.

Role in the Project:

Content Structuring: HTML is used to define the structure and layout of the weather app, including headings, paragraphs, lists, and other elements.

Embedding Media: HTML allows embedding of various media types, such as images and videos, which can be used to display weather maps, icons, and instructional videos.

Forms: HTML forms are used to capture user inputs, such as location selection and alert settings.
Key Elements Used:

<div>:
For general content division and layout structuring.

<header>, <footer>, <section>, <article>:
For semantic structuring of different parts of the app.

<form>:
For user input and location search.

<input>, <button>, <select>:
For interactive elements within forms.

<body>:
Contains the content of the HTML document, such as text, images, and other elements.

<h1> to <h6>:
Define headings, with <h1> being the highest level and <h6> the lowest

CSS (Cascading Style Sheets)

Overview:
CSS is used to control the presentation and layout of web pages. It allows developers to style HTML elements, making the web application visually appealing and user-friendly.

Role in the Project:
Styling:
CSS is used to apply styles to HTML elements, including colors, fonts, spacing, and layout.

Responsive Design:
CSS enables the creation of a responsive design that adapts to different screen sizes, ensuring the app is usable on both desktop and mobile devices.

Animations and Transitions:
CSS is used to add smooth animations and transitions, enhancing the user experience.
Key Features Used:

Selectors:
To apply styles to specific elements.

Box Model:
For layout and spacing (margin, border, padding, content).

Flexbox and Grid:
For creating flexible and responsive layouts.

Media Queries:
To implement responsive design.

JavaScript

Overview:
JavaScript is a versatile programming language used to create dynamic and interactive web applications. It allows developers to add interactivity and control the behavior of web pages.

Role in the Project:

Dynamic Content:
JavaScript is used to fetch and display real-time weather data from APIs.

User Interaction:
It handles user interactions, such as form submissions, button clicks, and location selection.

Data Processing:
JavaScript processes and formats the weather data before displaying it to the user.

Alerts and Notifications:
JavaScript is used to manage custom alerts and notifications set by users.
Key Concepts Used:

DOM Manipulation:
To dynamically update the content and structure of the web page.

Event Handling:
To respond to user actions like clicks and form submissions.

Fetch API / AJAX:
To retrieve weather data from third-party APIs without reloading the page.

Promises and Async/Await:
For handling asynchronous operations and ensuring smooth data fetching and processing.

Local Storage:
To store user preferences and settings on the client side.

Integration of Technologies

HTML, CSS, and JavaScript Integration:

HTML provides the structure of the web app.

CSS styles the HTML elements to create a visually appealing and responsive design.

JavaScript adds interactivity, fetches real-time data, and handles user interactions.

Example Workflow:

HTML defines the structure of the weather app, including sections for current weather, forecasts, and user settings.

CSS styles the app to ensure it is visually appealing and responsive across devices.

JavaScript fetches weather data from an API based on user input, processes the data, and dynamically updates the HTML to display the information.

Conclusion:

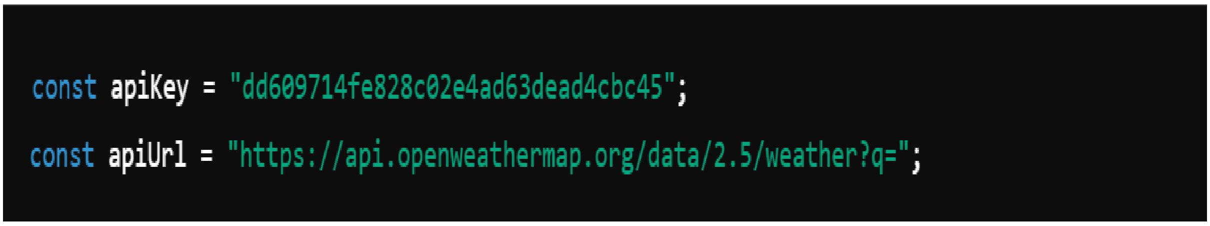
By using HTML, CSS, and JavaScript, the weather report web app achieves a structured, styled, and interactive user experience. HTML lays the foundation, CSS enhances the visual presentation, and JavaScript brings the app to life with dynamic data and user interactivity. This combination of technologies ensures that the weather app is both functional and user-friendly, providing users with a reliable and engaging tool to access weather information.

API Overview

The OpenWeatherAap API provides comprehensive weather data, including current conditions, forecasts, and historical data. It is accessible via HTTP requests and returns data in JSON format. The API requires an API key for access, ensuring secure and controlled usage.

API Key and URL:

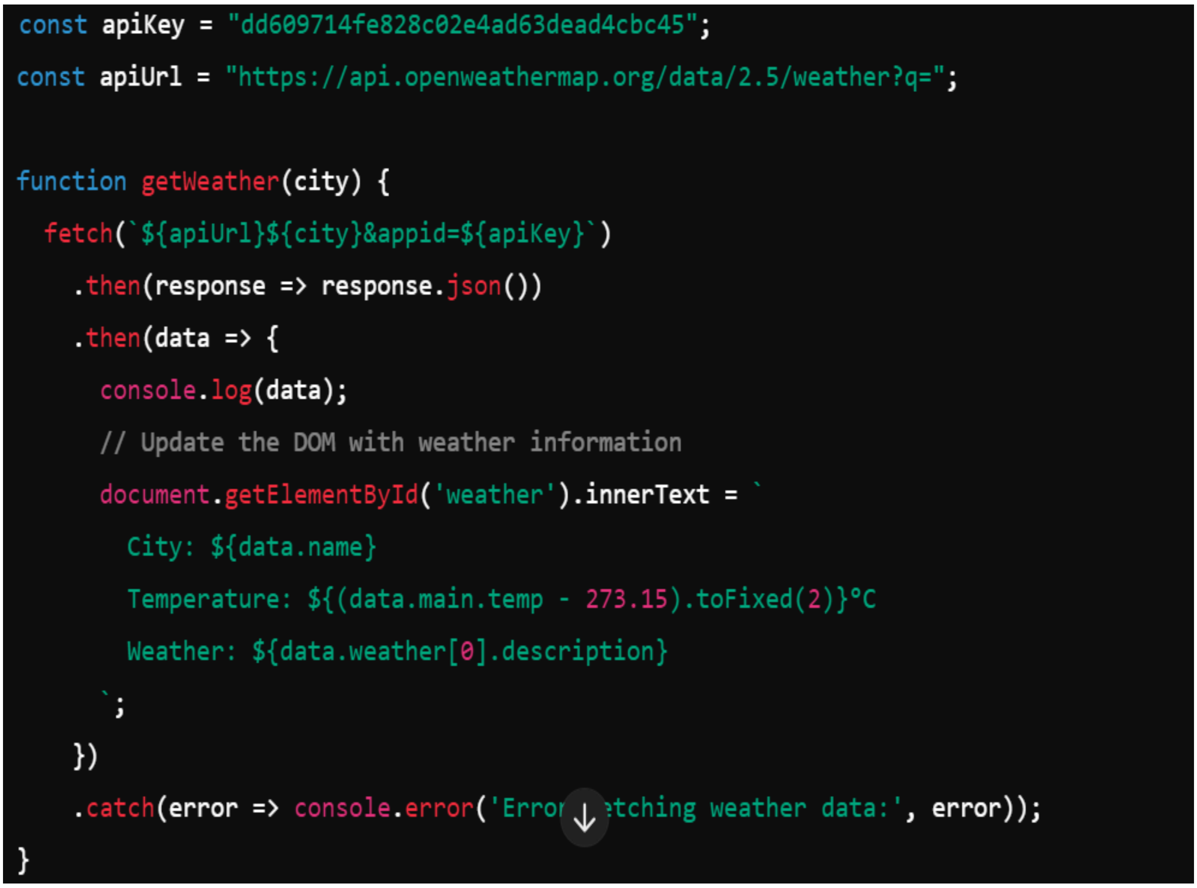
To use the OpenWeatherMap API, you need an API key and the appropriate endpoint URL. For our project, we used the following:



```
const apiKey = "dd609714fe828c02e4ad63dead4cbc45";
const apiUrl = "https://api.openweathermap.org/data/2.5/weather?q=";
```

API Integration

We integrated the OpenWeatherMap API into our project to fetch and display weather data. Below is an example of how we used JavaScript to make API requests and process the data:



```
const apiKey = "dd609714fe828c02e4ad63dead4cbc45";
const apiUrl = "https://api.openweathermap.org/data/2.5/weather?q=";

function getWeather(city) {
  fetch(`${apiUrl}${city}&appid=${apiKey}`)
    .then(response => response.json())
    .then(data => {
      console.log(data);
      // Update the DOM with weather information
      document.getElementById('weather').innerText = `
        City: ${data.name}
        Temperature: ${(data.main.temp - 273.15).toFixed(2)}°C
        Weather: ${data.weather[0].description}
      `;
    })
    .catch(error => console.error('Error fetching weather data:', error));
}
```

Results:

The integration of the OpenWeatherMap API enabled our website to display real-time weather information. Below are screenshots showing the weather data for different cities.

Challenges of API

Challenges:

- Handling API rate limits.
- Parsing and displaying data in a user-friendly manner.
- Ensuring compatibility across different browsers and devices.

SOLUTIONS OF API

Solutions:

- Implemented error handling to manage API request failures and rate limits.

- Used responsive design techniques to ensure the website looks good on all devices.

- Utilized JavaScript to dynamically update the UI based on the fetched weather data.

- Future Enhancements

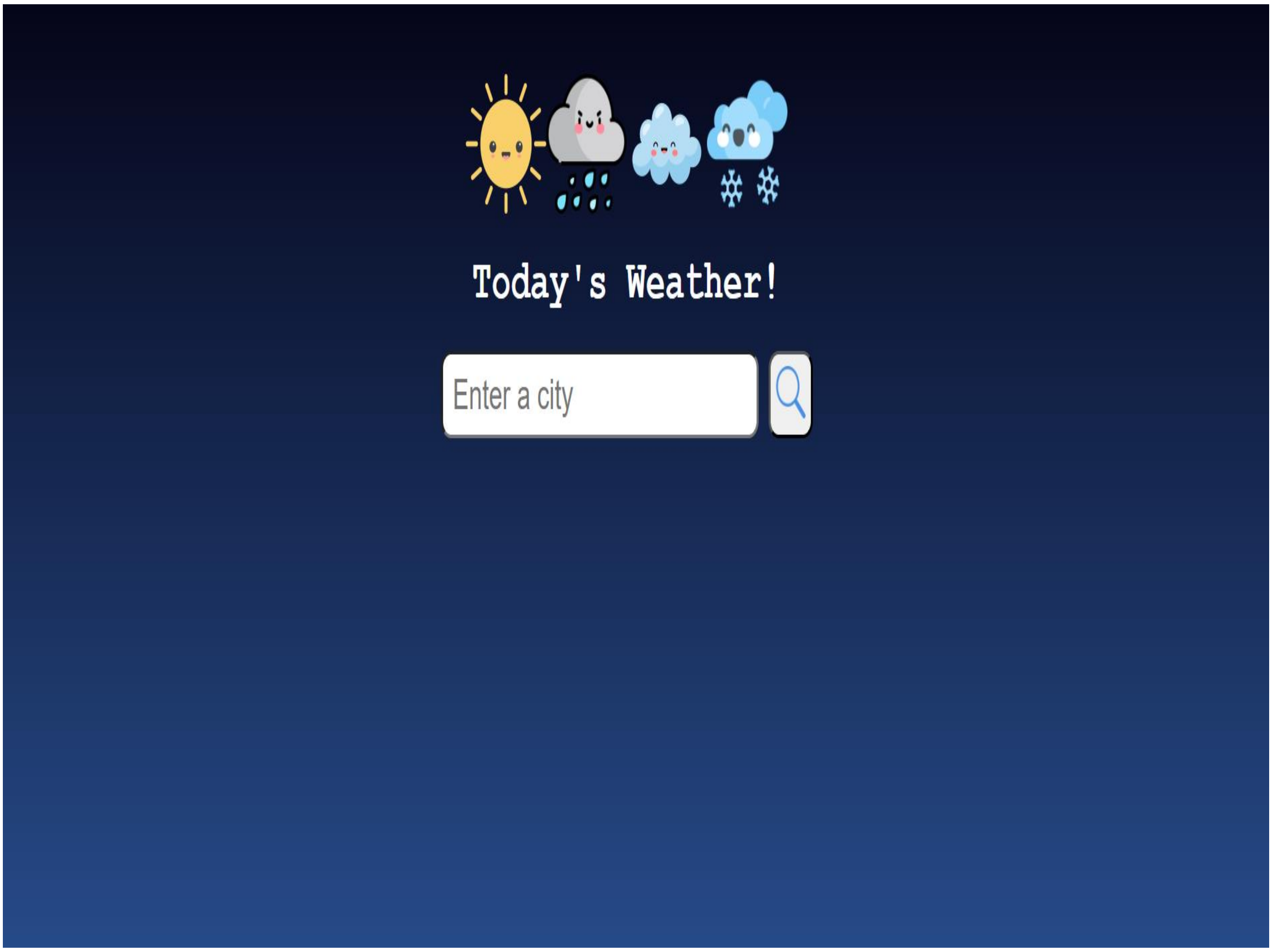
- Add features for hourly and weekly forecasts.

- Integrate additional weather-related data such as UV index and air quality.

- Implement user location detection to automatically show local weather.

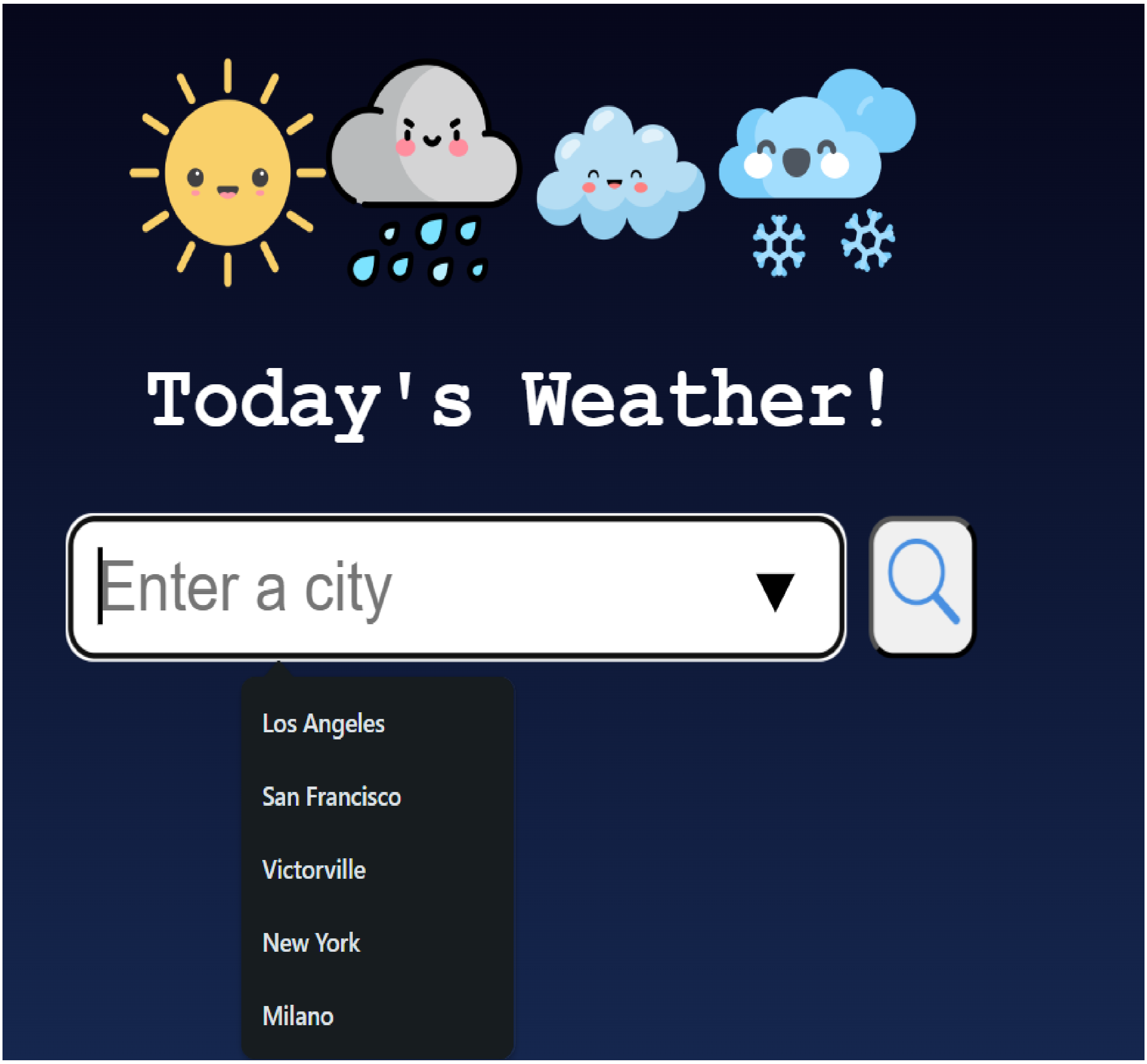
SCREENSHOTS OF WEB APP

FIRST VIEW OF WEB PAGE

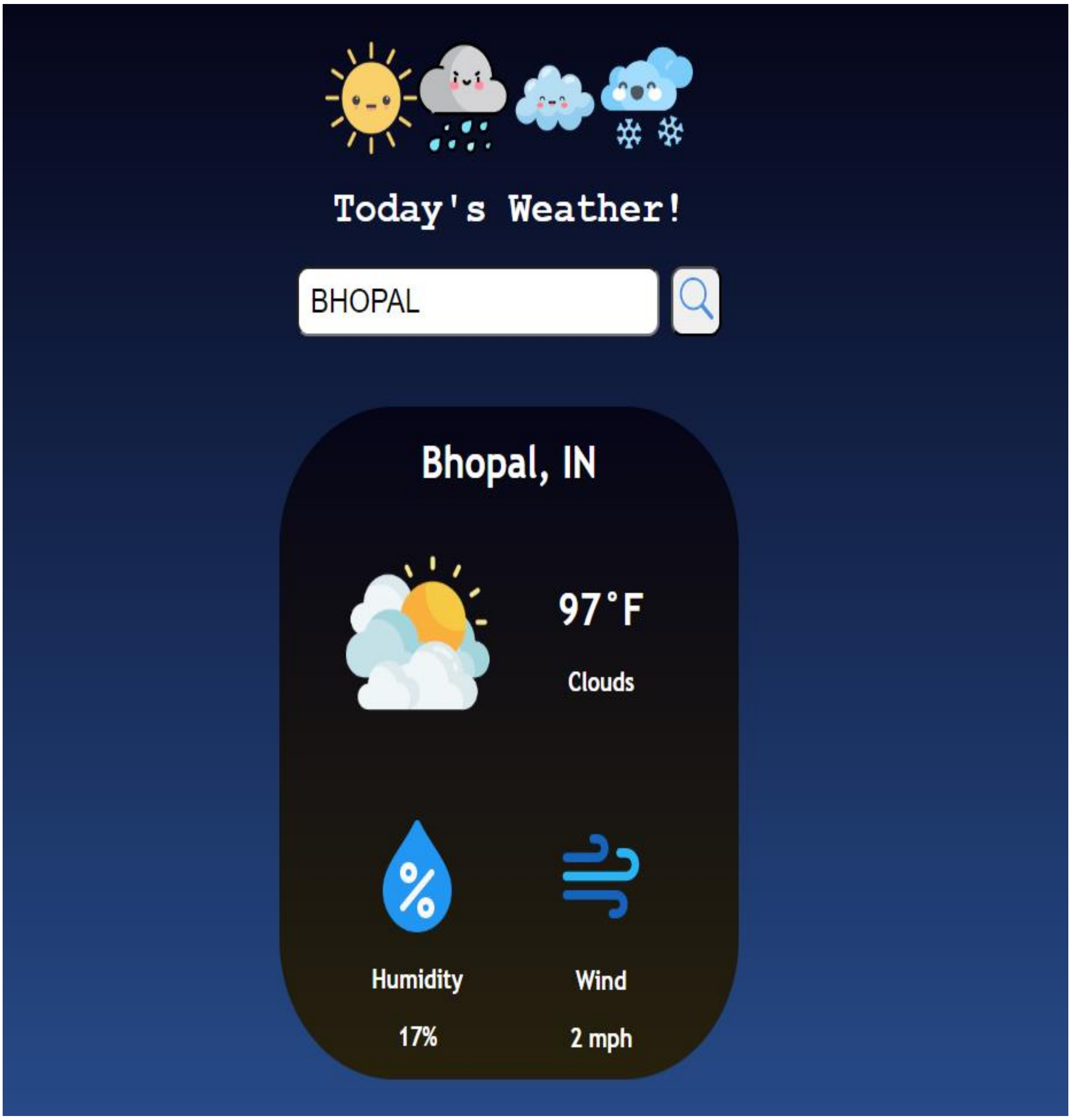


Today's Weather!

NOW WE HAVE TO TYPE THE CITY NAME



AFTER TYPEING THE CITY NAME CLICK ON THE SEARCH ICON



Today's Weather!

BHOPAL

Bhopal, IN

97°F

Clouds

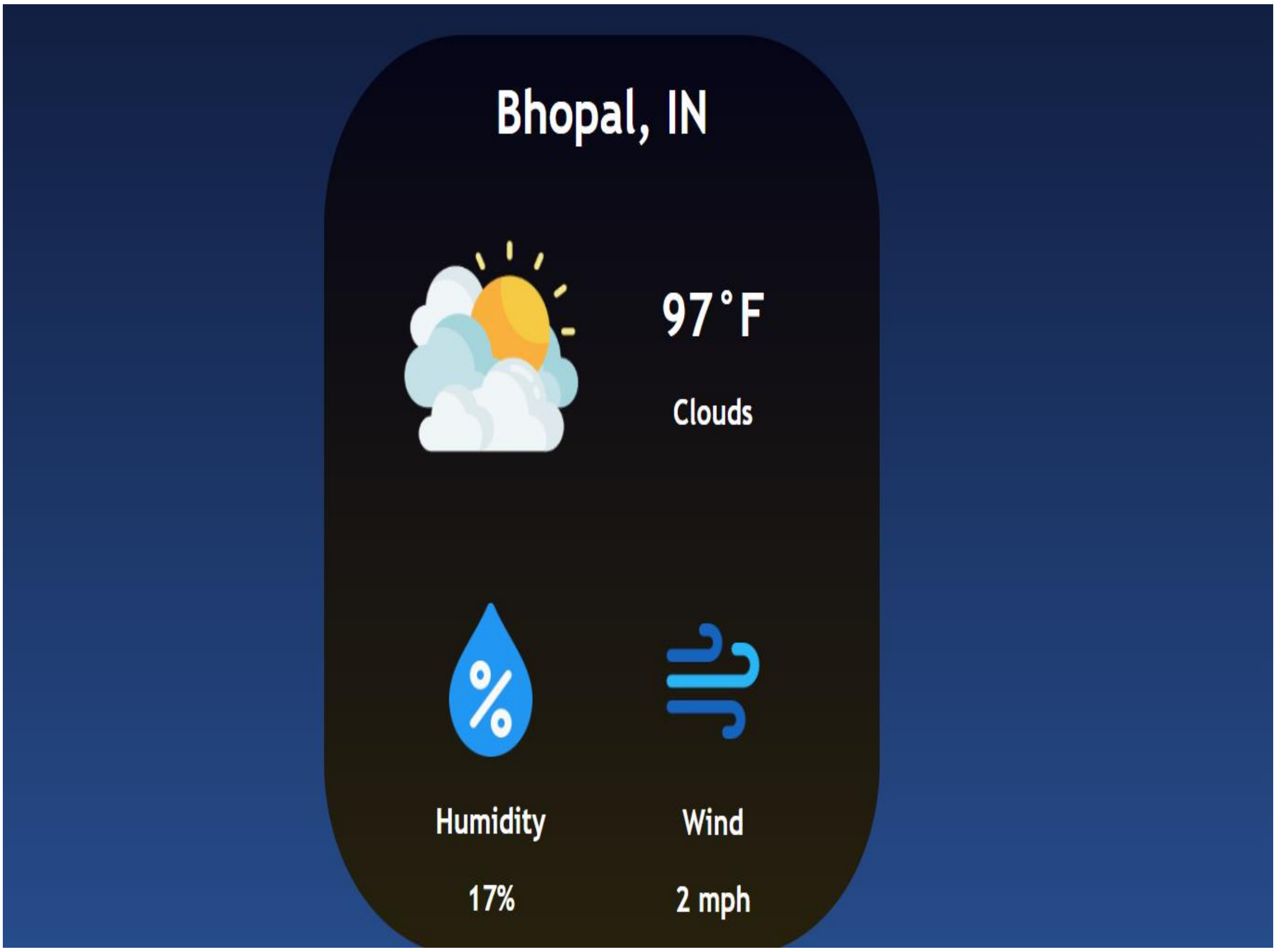
Humidity

17%

Wind

2 mph

THE FINAL RESULT IS THIS



Bhopal, IN

97˚F

Clouds

Humidity

17%

Wind

2 mph

Advantages of the Weather Forecasting APP

Developing a weather forecasting website offers numerous benefits both for the users and the developers. Here are some key advantages:

1. Real-Time Weather Updates

The website provides users with real-time weather information, ensuring they have access to the latest conditions and forecasts. This is crucial for planning daily activities and staying safe in adverse weather conditions.

2. Accessibility

The weather forecasting website is accessible from any device with internet connectivity, including smartphones, tablets, and computers. This ensures that users can check the weather anytime, anywhere.

3. User-Friendly Interface

A well-designed, user-friendly interface makes it easy for users to navigate the website and quickly find the weather information they need. Intuitive design enhances the user experience and increases engagement.

4. Customizable Features

Users can personalize their experience by setting favorite locations, choosing between different units of measurement (Celsius, Fahrenheit), and selecting specific weather data they want to view, such as temperature, humidity, wind speed, etc.

5. Comprehensive Weather Data

The website can provide detailed weather data, including current conditions, hourly forecasts, weekly forecasts, and additional information like UV index, air quality, and precipitation levels. This comprehensive data helps users make informed decisions.

6. Increased Safety

By providing accurate and timely weather forecasts, the website can help users stay safe during extreme weather events such as storms, hurricanes, or heatwaves. Alerts and warnings can be integrated to notify users of severe weather conditions.

7. Educational Value

The website can serve as an educational tool, offering insights into meteorology and weather patterns. Users can learn about different weather phenomena and how to interpret weather data.

8. Cost-Effective

For businesses and organizations, having a weather forecasting website can be more cost-effective than relying on third-party services. It allows for tailored features and functionalities specific to their needs.

9. Community Engagement

Local weather websites can foster community engagement by providing localized weather reports and involving the community in reporting weather conditions. This can be particularly useful for hyper-local weather information.

10. Integration with Other Services

Weather data can be integrated with other services and applications, such as travel planning, event management, agriculture, and outdoor activities. This makes the website a valuable tool for a wide range of users and industries.

APPLICATIONS

A weather app can have a wide range of applications, catering to different user needs and industries. Here are some common applications of a weather app:

1.Daily Weather Updates

General Users:
Provide daily weather forecasts to help users plan their day.
This includes temperature, precipitation, wind speed, and humidity information.

2. Travel Planning

Tourists and Travelers:
Offer weather forecasts for travel destinations to help users pack appropriately and plan outdoor activities.

3. Outdoor Activities

Sports Enthusiasts:
Deliver weather updates for planning sports activities such as hiking, biking, skiing, and other outdoor sports.

Event Planners:
Assist in organizing outdoor events like weddings, festivals, and concerts by providing accurate weather forecasts.

4. Agriculture

Farmers:
Supply weather information critical for agricultural activities such as planting, irrigation, and harvesting. This includes forecasts for rain, frost, and extreme temperatures.

Gardening:
Help home gardeners manage their plants and garden activities by providing local weather forecasts.

5. Disaster Preparedness

Emergency Services:

Aid in disaster preparedness by providing early warnings for severe weather conditions like storms, hurricanes, and floods.

Public Safety:

Inform the public about imminent severe weather conditions, helping them take necessary precautions.

6. Health and Wellness

Health Alerts:

Provide information on weather-related health alerts such as high UV index, extreme heat, or cold conditions which can impact health.

Air Quality:

Include air quality index (AQI) information to help users with respiratory issues manage their outdoor activities.

7. Transportation

Aviation:

Supply pilots and airlines with weather updates crucial for flight planning and safety.

Shipping:

Assist maritime shipping companies with weather information to ensure safe navigation and scheduling.

Road Transportation:

Provide drivers with weather forecasts for safe travel and route planning.

8. Home Automation

Smart Homes:

Integrate with smart home systems to adjust heating, cooling, and lighting based on weather conditions.

9. Energy Management

Renewable Energy:
Aid solar and wind energy producers by providing weather forecasts to optimize energy production and storage.

Utilities:
Help utility companies forecast energy demand based on weather patterns.

10. Business Operations

Retail:
Assist retail businesses in managing inventory and sales strategies based on weather forecasts (e.g., stocking seasonal clothing).

Construction:
Provide construction companies with weather forecasts to plan and manage construction activities safely and efficiently.

11. Education and Research

Schools and Universities:
Offer weather information for educational purposes and field trip planning.

Researchers:
Provide accurate weather data for environmental and meteorological research.

12. Leisure and Lifestyle

Leisure Activities:
Assist users in planning leisure activities such as beach outings, picnics, and barbecues.

Fashion:
Offer fashion advice based on current and forecasted weather conditions.

By incorporating these applications, a weather app can serve a diverse audience, offering tailored weather information to meet specific needs and enhance daily life.

Future Work

The weather app can be further enhanced and expanded in several ways to increase its functionality, user engagement, and overall utility. Here are some potential areas for future development:

1. Enhanced Forecasting Features

Hourly Forecasts*:*

Add detailed hourly weather forecasts for the next 24 hours to provide users with more precise short-term weather information.

Extended Forecasts: Include 10-day or even 14-day forecasts to help users plan further ahead.

2. Advanced Weather Data

Weather Alerts*:*

Integrate severe weather alerts and notifications to keep users informed about extreme conditions such as storms, hurricanes, or heatwaves.

Air Quality Index (AQI):

Provide information on air quality, including levels of pollutants and health advisories.

UV Index:

Display UV index readings to help users take precautions against harmful sun exposure.

Pollen Count: *Include pollen count data to assist users with allergies in managing their activities.*

3. User Personalization

User Accounts*:*

Implement user accounts to allow users to save favorite locations, set preferences, and receive customized weather updates.

Personalized Notifications:

Offer personalized weather notifications based on user preferences, such as morning summaries or alerts for specific weather conditions.

4. Improved User Interface and Experience

Interactive Maps*:*
Integrate interactive weather maps showing radar, satellite imagery, and weather patterns.

Dynamic Backgrounds*:*
Change the app's background based on current weather conditions to provide a more immersive experience.

Localization*:*
Support multiple languages and regional units (e.g., Celsius/Fahrenheit, kilometers/miles) to cater to a global audience.

5. Integration with Other Services and Devices

Smart Home Integration:
Connect with smart home devices to adjust home settings (like thermostat, lighting) based on the weather.

Calendar Integration:
Sync with users' calendars to provide weather updates for planned events and activities.

6. Data Analysis and Insights

Historical Weather Data*:*
Provide access to historical weather data for users interested in past weather trends.

Weather Insights*:*
Offer insights and recommendations based on weather patterns, such as best times for outdoor activities or travel.

7. Social and Community Features

User Reports*:*
Allow users to report local weather conditions, enhancing real-time data accuracy.

Community Sharing*:*
Enable users to share weather updates and forecasts on social media platforms.

8. Technical Enhancements

Performance Optimization*:*
Continuously optimize the app’s performance to ensure fast loading times and smooth operation.

Offline Mode:
Develop an offline mode that allows users to access the last updated weather information without an internet connection.

Security Enhancements:
Implement advanced security measures to protect user data and ensure privacy.

By implementing these future enhancements, the weather app can become more versatile, user-friendly, and valuable to a wider audience. These improvements will not only enrich the user experience but also broaden the app’s appeal and functionality.

CONCLUSION

The weather app project has been instrumental in applying fundamental web development skills to produce a practical and effective application. By integrating HTML, CSS, and JavaScript with the OpenWeatherMap API, we successfully delivered a tool that offers real-time weather updates and forecasts to users. This project underscores the power of technology in addressing everyday needs, enhancing user convenience, and fostering a deeper understanding of API integration and data visualization.

Through this endeavor, valuable lessons in project planning, iterative development, and user-centric design have been learned. These insights not only bolster technical proficiency but also underscore the importance of responsive design and seamless user experience across different devices. Moving forward, the project's foundation serves as a launchpad for future innovations, encouraging exploration into advanced features like personalized notifications, enhanced data analytics, and integration with emerging technologies. Ultimately, this project exemplifies how web development can tangibly improve daily life and sets the stage for continued growth in software development endeavors.

Source Code

HTML

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" href="wstyles.css">
        <script src="wstyle.js"></script>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Weather</title>
    </head>
    <body>
        <div class="everything">
        <div class="search">
            <div class="images">
            <img id="logo" src="./cartoon-sun-icon-with-facial-expression-free-png.webp">
            <img id="logo" src="./2204339.png">
            <img id="logo" src="./cartoon-cloud-icon-png.webp">
            <img id="logo" src="./2323367.png">
            </div>
            <h1>Today's Weather!</h1>
            <div class="city">
                <form>
                    <input id="enter" placeholder="Enter a city" list="browsers" name="browser">
                    <datalist id="browsers">
                      <option value="Los Angeles">
                      <option value="San Francisco">
                      <option value="Victorville">
                      <option value="New York">
                      <option value="Milano">
                    </datalist>
                  </form>
            <button id="searchButton" onclick="search()"><img
src="./search-icon-sl7.png"></button>
            </div>
    </div>
    <div class="weather">
        <div class="loc">
        <h1 class="location"></h1>
        </div>
        <div class="columns">
        <div class="col1">
            <div>
            <img id="today-weather" src="./weather01-512.webp">
            </div>
            <div class="add">
                <img id="logod" src="./728093.png">
                <h3>Humidity</h3>
                <h3 class="humid"></h3>
            </div>
        </div>
        <div class="col2">
            <div class="new">
                <h1 class="temp"></h1>
                <h3 class="description"></h3>
                </div>
        <div class="add">
                <img id="logod" src="./3944594.png">
                <h3>Wind</h3>
                <h3 class="speed"></h3>
        </div>
        </div>
        </div>
    </div>
    </div>
    </body>
```

Source Code

CSS

```
body, html{
    padding: 0;
    margin: 0;
}
body{
    background-size: auto 100%;
    background-image: linear-gradient(rgb(5, 5, 24),rgb(39, 76, 139));
    width: 100vw;
    display: flex;
    justify-content: center;
    min-height: 100vh;
}

.search{
    display: flex;
    height: 100%;
    width: 100%;
    align-items: center;
    flex-direction: column;
}
.everything{
    display: flex;
    flex-direction: column;
    height: 100%;
    width: 100%;
    margin: 0 auto;
    position: relative;
    align-items: center;
}
.city{
    display: flex;
    justify-content: center;
    align-items: center;
    width: 70%;
    margin: 0 auto;
    flex-direction: row;
    margin-bottom: 50px;
}
.images{
    display: flex;
    width: 50%;
    align-items: center;
    justify-content: center;
    margin-top: 40px;
}
h1, h2, h3, h4{
    color: white;
    font-family:'Trebuchet MS', 'Lucida Sans Unicode', 'Lucida
Grande', 'Lucida Sans', Arial, sans-serif;
}
.search h1{
    font-family: 'Courier New', Courier, monospace;
}

#logo{
    width: 80px;
    height: 80px;
}
#logod{
    flex: 1;
    margin-top: 50px;
    width: 50%;
    height: 50%;
}
#today-weather{
    width: 100%;
    height: 100%;
}
```

```
.new{
   height: 100%;
   width: 100%;
   flex: 1;
   padding-top: 25px;
   padding-bottom: 20px;
}

input{
   box-sizing: border-box;
   height: 50px;
   width: 100%;
   font-size: x-large;
   padding: 5px 10px;
   border-radius: 10px;
}
.search button{
   padding: 5px 5px;
   height: 50px;
   margin-left: 10px;
   border-radius: 10px;
}
.search button:hover{
   background-color: rgb(172, 168, 168);
}
.search button img{
   width: 30px;
   height: 30px;
}
.weather{
   align-items:center;
   justify-content: space-evenly;
   flex-direction: column;
   display: none;
   background-image: linear-gradient(rgb(5, 5, 24),rgb(39, 33, 13));
   width: 300px;
   height: 100%;
```

```
    border-radius: 100px;
    min-width: 400px;

}
datalist {
    position: absolute;
    background-color: white;
    border: 1px solid blue;
    border-radius: 0 0 5px 5px;
    border-top: none;
    font-family: sans-serif;
    width: 350px;
    padding: 5px;

  }
  option {
    background-color: white;
    padding: 4px;
    color: blue;
    margin-bottom: 1px;
     font-size: 18px;
    cursor: pointer;
  }
.loc{
 align-items: center;
 justify-content: center;
 height: 10%;
 margin-bottom: 0;
 display: flex;
}
.add{
    text-align: center;
    align-items:center;
    justify-content: flex-end;
    flex-direction: column;
    width: 100%;
    flex: 1;
    height: 100%;
    margin: auto;
}
.col1{
    max-width: 500px;
    justify-content: space-around;
    flex: 1;
    margin: 0 auto;
    width: 50%;
    height: 100%;
    float: left;
}
.col2{
    flex: 1;
    max-width: 500px;
    margin: 0;
    width: 50%;
    height: 100%;
    float: right;
}
.columns{
    display: flex;
    flex-direction: row;
    width: 80%;
    height: 100%;
    align-items: center;
    margin: 0 auto;
    text-align: center;
}
@-ms-viewport{
    width: device-width;
  }
```

Source Code

JAVA-SCRIPT

```
const apiKey = "dd609714fe828c02e4ad63dead4cbc45";
const apiUrl = "https://api.openweathermap.org/data/2.5/weather?q=";



async function checkWeather(city){
    const response = await fetch(apiUrl + city + "&units=imperial" +
`&appid=${apiKey}`);

    var data = await response.json();
    console.log(data);

    document.querySelector(".location").innerHTML = data.name + ",
" + data.sys.country;

    document.querySelector(".speed").innerHTML =
Math.round(data.wind.speed) + " mph";

    document.querySelector(".humid").innerHTML =
data.main.humidity + "%";

    document.querySelector(".temp").innerHTML =
Math.round(data.main.temp) + "&deg;F";

    document.querySelector(".description").innerHTML =
data.weather[0].main;
```

```
    if (data.weather[0].main == "Clouds"){
        document.getElementById("today-weather").src = "./clouds.png"
    }
    else if(data.weather[0].main == "Clear"){
        document.getElementById("today-weather").src = "./clear.png"
    }
    else if(data.weather[0].main == "Drizzle"){
        document.getElementById("today-weather").src =
"./drizzle.png"
    }
    else if(data.weather[0].main == "Mist"){
        document.getElementById("today-weather").src = "./mist.png"
    }
    else if(data.weather[0].main == "Rain"){
        document.getElementById("today-weather").src =
"./4837776.png"
    }
    else if(data.weather[0].main == "Snow"){
        document.getElementById("today-weather").src = "./snow.png"
    }
    document.querySelector(".weather").style.display = "flex";
}



function search(){
    checkWeather(document.getElementById('enter').value);
}
```

SCAN AND VISIT OUR WEBSITE

THANK YOU