National Research Journal of Business Economics
Volume No: 13, Issue No: 1, Year: 2026 (January-June)
PP: 147-151
ISSN: 2349-2015
Peer Reviewed & Refereed Journal (IF: 6.74)
Journal Website: www.nrjbe.in

A STUDY ON THE CREATION OF DYNAMIC WEATHER WEBSITE (USING CODE: HTML, CSS, JAVA SCRIPT)

Revathi G

Assistant Professor
Department of Commerce, Rathinam College of Arts and Science, Tamil Nadu

Navaneethan S

II M.com CA
Department of Commerce, Rathinam College of Arts and Science, Tamil Nadu

ABSTRACT

Weather describes the dynamic atmospheric conditions at specific times and locations, including temperature, humidity, precipitation, wind, and pressure. These conditions are influenced by solar radiation, latitude, altitude, water proximity, and topography, manifesting as rain, snow, storms, fog, and other phenomena. Meteorologists measure these using instruments like thermometers, barometers, and anemometers. Weather significantly impacts agriculture, transportation, tourism, and public health, with extreme events causing damage and disruption.

This project created "Clime," a dynamic weather website using HTML, CSS, and JavaScript. The site provides an attractive, accessible interface displaying real-time weather information including temperature, humidity, and wind speed for user-specified locations. HTML structures the content, CSS handles styling, and JavaScript enables dynamic functionality. The website integrates with the OpenWeatherMap API to fetch current weather data, ensuring accurate, up-to-date forecasts. A robust search function allows users to easily look up conditions for any location, delivering relevant results with weather descriptions and atmospheric details.

This project report aims to share insights into effective web development practices, highlighting the successful integration of front-end technologies to produce a robust and user-focused application.

KEYWORDS: Dynamic Weather Website, Real-time Weather Data, Weather Forecasting OpenWeatherMap API, Temperature, User Interface (UI), Front-end Web Development.

INTRODUCTION

Clime is a dynamic weather website built using HTML, CSS, and JavaScript. It pulls real-time data from the OpenWeatherMap API to display current weather conditions — including temperature, humidity, and wind speed — for any location a user searches.

The project sits within the broader field of front-end web development, where HTML structures the content, CSS handles the visual presentation, and JavaScript drives the interactive behavior. Visual Studio Code was used as the development environment throughout.

The core goals of Clime are simple: give users accurate, live weather data through a clean and attractive interface, with a reliable search function that works for locations worldwide. When a valid city is entered, weather details animate smoothly into view. When a location isn't found, a friendly error page is shown instead.

ISSN: 2349-2015
Peer Reviewed & Refereed Journal (IF: 6.74)
Journal Website: www.nrjbe.in

Objectives

1. To provide real-time weather information
2. To display key details like temperature, humidity, and wind speed
3. To create a user-friendly and attractive interface for easy access
4. To allow users to search weather details for different location

TECHNOLOGY USED

1. HTML

HTML is used to create the structure of the website. It defines elements like headings, input fields, buttons, and containers to display weather data.

2. CSS

CSS is used to design and style the website. It improves the appearance by adding colors, layouts, fonts, and responsiveness.

3. JavaScript

- JavaScript is used to make the website dynamic.
- It Fetches data from the weather API
- Updates the UI dynamically

SYSTEM TESTING

System testing is a level of software testing where the complete, integrated application is tested as a whole to verify it meets specified requirements

- **Black Box** Testing verified that all user-facing functionality — searching cities, displaying weather data, handling errors — behaved as expected without examining internal code logic.
- **White Box** Testing examined the internal JavaScript logic, particularly the API fetch chain, switch-case condition mapping, and DOM manipulation to ensure code efficiency and correctness.
- **Regression Testing** validated that edge cases such as empty inputs, invalid city names, and network responses with error codes (404) were handled gracefully without application crashes.
- **Unit Testing** confirmed that individual components — the search handler, the weather renderer, and the error panel toggler — each functioned independently and correctly before integration.
- **Acceptance Testing** evaluated the overall user experience to confirm the application met its original objectives and was ready for deployment.

SYSTEM IMPLEMENTATION

System implementation is the process of installing, configuring, and deploying a fully developed system into a live/production environment so end-users can begin using it

- **Structure** — HTML defined the container, search box, weather display, and error sections

ISSN: 2349-2015
Peer Reviewed & Refereed Journal (IF: 6.74)
Journal Website: www.nrjbe.in

- **Styling** — CSS applied the glassmorphism theme, animations, and responsive layout
- **Logic** — JavaScript wired the search button to the OpenWeatherMap API and rendered results dynamically
- **Testing** — Multiple testing phases were conducted to validate correctness and usability
- **Deployment** — The final application was made accessible via a browser with no server-side dependency

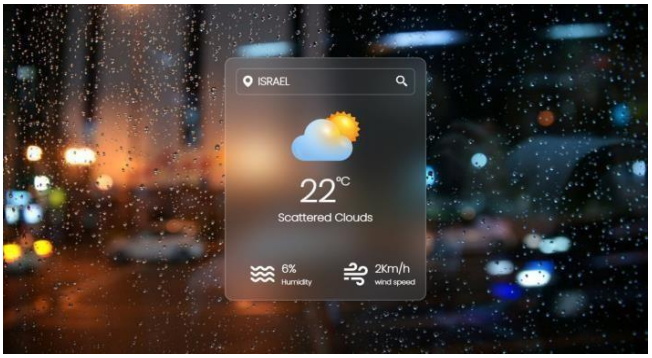
RESULT

The screenshots from the project demonstrate successful rendering across a variety of weather conditions:

- **Scattered Clouds** — Israel: 22°C, 6% humidity, 2 km/h wind
- **Light Rain** — USA: 26°C, 58% humidity, 0 km/h wind
- **Clear Sky** — New York: -1°C, 70% humidity, 2 km/h wind
- **Mist** — Lucerne: 3°C, 93% humidity, 10 km/h wind
- **Error State** — Invalid input triggers the 404 not-found panel

SCREENSHOTS

WEATHER INFORMATION PAGE



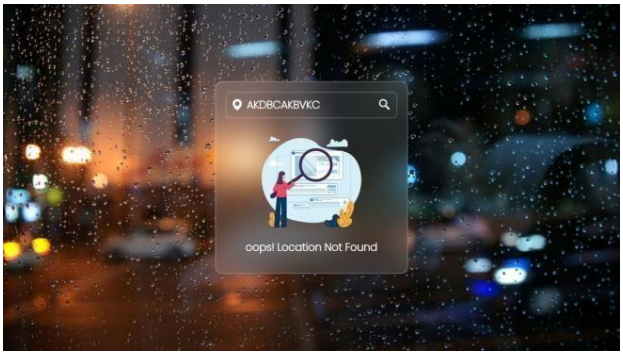
LIGHT RAIN CONDITION

ISSN: 2349-2015
Peer Reviewed & Refereed Journal (IF: 6.74)
Journal Website: www.nrjbe.in

CLEAR SKY CONDITION

MIST CONDITION

LOCATION NOT FOUND ERROR PAGE



CONCLUSION

The development of Clime represents a significant demonstration of how modern front-end web technologies can be harnessed to build practical, real-world applications that serve everyday user needs. By integrating HTML, CSS, and JavaScript with the OpenWeatherMap API, this project successfully delivers a seamless and responsive weather browsing experience without the need for any server-side infrastructure or complex back-end architecture. The application achieves all of its core objectives — providing real-time weather data, displaying

ISSN: 2349-2015
Peer Reviewed & Refereed Journal (IF: 6.74)
Journal Website: www.nrjbe.in

essential metrics such as temperature, humidity, and wind speed, implementing a reliable and accurate location search function, and presenting the information through a visually appealing glassmorphism interface. Throughout the development process, careful attention was given to user experience design, ensuring that animations, transitions, and error handling all contribute to a smooth and intuitive interaction. The rigorous testing phases, covering black box, white box, regression, unit, and acceptance testing, further validated the reliability and correctness of the application across a wide range of real-world weather conditions and location inputs, confirming that Clime is both functionally sound and user-ready.Looking ahead, Clime lays a strong and scalable foundation upon which future enhancements can be built to further enrich the user experience and expand the application's capabilities. Planned improvements include the integration of a 5-day extended weather forecast, automatic location detection using the browser's built-in Geolocation API, and a unit toggle feature allowing users to switch between Celsius and Fahrenheit based on their preference. Additional features such as hourly weather breakdowns, air quality index display, UV index monitoring, and dynamic background changes based on real-time weather conditions could further elevate the application into a comprehensive weather platform. As web technologies continue to evolve rapidly, Clime also presents opportunities for progressive enhancement — including converting it into a Progressive Web App (PWA) to enable offline access and mobile installation. Ultimately, this project not only demonstrates the power and versatility of front-end development but also underscores the importance of thoughtful design, clean code practices, and continuous improvement in building digital tools that are both meaningful and impactful to users across the globe.

REFERENCE

1. OpenWeatherMap. (2024). Weather API Documentation. Retrieved from:
2. https://openweathermap.org/Api
3. Mozilla Developer Network (MDN). (2024). Web Development Documentation
4. (HTML, CSS, JavaScript). Retrieved from: https://developer.mozilla.org
5. Duckett, J. (2011). HTML and CSS: Design and Build Websites. Wiley Publishing.
6. Flanagan, D. (2020). JavaScript: The Definitive Guide (7th Edition). O'Reilly Media.
7. Visual Studio Code Documentation. (2024). Code Editor Guide. Retrieved from:
8. https://code.visualstudio.com/docs