International Journal of Data Informatics and Intelligent Computing (IJDIIC)
Vol. 4, No. 3, September 2025, pp. 1~13
DOI: 10.59461/ijdiic.v4i3.199

An Improved Web-Based Weather Information Retrieval Application

Abba Almu[1], Abdulkudus Yunusa[1]
[1]Department of Computer Science, Usmanu Danfodiyo University, P.M.B 2346, Sokoto, Nigeria.

Article Info

Article history:

Received April 02, 2025
Revised June 22, 2025
Accepted July 01, 2025

Keywords:

Weather
Information Retrieval
Multilingual Support
Open Weather Map API
Caching Mechanism
Translation Mechanism

ABSTRACT

The Improved Web-Based Weather Information Retrieval Application was implemented to address Nigeria's challenges in weather information retrieval, including limited data collection and accessibility issues for diverse linguistic groups. The system integrates advanced forecasting models, real-time updates, caching mechanisms, and multilingual support for Hausa, Yoruba, and Igbo, ensuring inclusivity and accessibility, particularly for rural users. With a user-friendly interface, the application caters for users with varying technical expertise and supports critical sectors like agriculture and disaster management. The results of the system evaluation revealed that the existing system took a time of 1.774s at the speed of 8.719s to retrieve relevant weather information, while the proposed system took a lesser time of 0.753s at a faster speed of 3.929s to retrieve relevant weather information. This result shows significant improvements over the existing system, which lacked caching mechanisms and multilingual support, resulting in slower data retrieval and limited accessibility. It also demonstrated its effectiveness in enhancing decision-making, climate resilience, and disaster preparedness.

Corresponding Author: Abba Almu (e-mail: almu.abba@udusok.edu.ng)

1. INTRODUCTION

Weather forecasting information plays a vital role in helping individuals, businesses, and governments make informed decisions. Accurate weather information can significantly impact sectors such as agriculture, transportation, disaster management, and public safety. In Nigeria, a country with diverse weather patterns ranging from arid regions in the north to tropical rainforests in the south, the need for reliable weather forecasting is crucial. The effects of climate change [1] further emphasize the importance of accessible, accurate, and timely weather information. Inadequate weather forecasting affects several critical sectors in Nigeria. Agriculture, which is a key pillar of the Nigerian economy, relies heavily on accurate weather information for planning planting and harvesting seasons, managing irrigation, and preparing for adverse weather conditions. Poor weather forecasts can lead to crop failures, pest infestations, and reduced agricultural productivity, directly impacting food security and the livelihoods of millions of Nigerians. Additionally, Nigeria frequently experiences extreme weather events, such as floods and droughts, which have devastating consequences for communities and infrastructure. Inaccurate or delayed weather information impairs disaster preparedness and response, increasing the risk to human life and property.

Weather forecasting in Nigeria faces several significant challenges affecting accuracy, accessibility, and effectiveness. These issues impact various sectors, including agriculture, disaster management, transportation, public health, and socio-economic activities. The study by [2] developed a web-based application designed to provide accurate weather information tailored to users' locations. The application utilizes the Open Weather Map API to retrieve weather data specific to the user's location. The data is presented in a user-friendly web interface that is responsive and adapts to various devices, ensuring accessibility for all users. Despite this effort, the study possessed the following problems: The absence of a caching mechanism in the existing system leads to frequent API calls, causing higher server costs, potential rate-limit issues and ineffective user experience due to slower data retrieval and reliance on live data for

every request and it also provides weather information using only English language which is difficult for those users that do not understand English.

To solve these problems, this study developed an improved web-based weather forecasting application that enhances the accuracy and accessibility of weather information in Nigeria. The study integrated a caching mechanism to enhance the system performance by minimizing redundant API calls and ensuring faster data retrieval. It also enabled the proposed application to cater for three major Nigerian languages consisting of Hausa, Yoruba, and Igbo.

2. LITERATURE REVIEW

The study [3] developed a weather application using Android Studio aimed at simplifying weather forecasting for users, particularly farmers who depend on weather conditions for their crops. The application provides essential weather details such as temperature, humidity, and rain probability with just a click, requiring only an internet connection and a Global positioning system (GPS). The application was designed with a user-friendly interface, making it accessible even for illiterate users. It allows for quick access to vital weather information without involving complex steps. However, the application depends on a stable internet connection and is restricted to users with GPS-enabled devices; this can be a significant drawback, especially in rural areas where internet access may be inconsistent or unavailable.

The researcher [4] proposed a machine learning approach for improving weather forecasting accuracy with reduced resource requirements for rainfall and flooding predictions in urban areas. The study used historical meteorological data from 2009–2023 from India. It then applied supervised learning models consisting of regression and ensemble methods. It also processed the data by handling missing values, outlier detection, and normalization. The models also evaluated accuracy metrics to determine prediction accuracy. The results indicated that the proposed models provide significant accuracy improvements with 80.11%. However, the data used for the experiments are too small compared to the volume of data required for the algorithm to be trained well.

In [5] presented a weather monitoring and reporting system which has the potential to improve the accuracy and reliability of weather data by using IoT technology. The system monitored the collection of real-time data on various weather conditions consisting of temperature and humidity with the help of sensors and other IoT devices. The collected data are then analyzed and used to generate accurate weather reports that the organizations can use for various purposes. The results demonstrated that the proposed IoT-based weather monitoring system can provide valuable insights and information that can help people better understand and prepare for the weather. However, the study does not provide other weather-related information, such as humidity, wind speed, or forecasts, which could be valuable to users, and the application is unable to provide weather information for distant locations.

The study [6] presented a weather forecasting application developed using the Python Django framework. It focuses on providing accurate and timely weather information through a comparative study of various APIs for weather data retrieval. The application aims to deliver current weather conditions, temperature variations, and detailed forecasts for the upcoming six hours, ensuring users have access to reliable information at all times. The application is designed with a focus on user experience, making it easy for users to access weather information anytime and anywhere. It also provides timely updates on weather conditions, adapting its interface to reflect real-time changes, which is crucial for users needing current information. However, the application is unable to meet the needs of users requiring longer-term weather predictions because it only provides forecasts for the next six hours, and its accuracy is also unable to be ensured if the selected APIs supply inaccurate data.

The authors in [7] developed a mobile application that integrates augmented reality (AR) with weather forecasting. The application aims to provide users with a more engaging and informative way to understand local weather conditions by visualizing weather data in 3D. It utilizes a weather forecasting API and data collected from sensors like the DHT11, which measures temperature and humidity, to deliver real-time weather updates. The integration of AR provides a unique and immersive experience, making it easier for users to understand complex weather data through visual representation. The application offers real-time weather updates, which are crucial for users needing timely information for planning their activities. However, the application is unable to provide accurate weather data due to the limitations in precision and range of DHT11 sensors, and some users may be unable to use the AR features without compatible devices or the necessary technical knowledge.

The study [8] presented Weather Forecasting Using Artificial Neural Networks in India. It identified weather conditions such as Minimum temperature, Precipitation Rate, wind speed, Maximum temperature and pressure because of their many important roles in the field of cultivation. This is due to the fact that a country like India, knowing its precise weather forecasting, will enable farmers to achieve better productivity of crops. This study also employed a technique that is suited for processing the nonlinear conditions of weather and making a prognosis. The results of the study revealed that the combination of MLFFNN with the

BPA algorithm proved to be better for weather prediction. However, the application is unable to provide accurate predictions if the Artificial Neural Network (ANN) models rely on meteorological data that is of poor quality or lacks comprehensiveness.

The authors in [9] developed an online Website that displays real-time Weather Updates of different places. The Website is developed by using some basic frontend Web Development Technologies such as HTML, CSS, JavaScript and API Technology for displaying real-time Data. It provides accurate and timely weather information in a dynamic environment. The results of the system testing demonstrated that the users are satisfied with a convenient platform for accurate weather forecasts and its accessibility across devices. However, the application is unable to provide comprehensive real-time data for all geographical areas, limiting its usefulness in remote or less populated regions.

In [10] presented a real-time weather forecasting model designed for disaster management, utilizing machine learning algorithms to enhance prediction accuracy. The application is developed to be user-friendly, allowing individuals without prior knowledge of weather calculations to understand forecasts through intuitive logos and visual representations. It utilizes data from the Andhra Pradesh State Disaster Management Authority (APSDMA) and employs an open weather API for real-time data retrieval. Machine learning algorithms, specifically Artificial Neural Networks (ANN) and Gaussian Process Regression (GPR), are implemented for accurate predictions. The accuracy of the predictions is heavily reliant on the quality and availability of data from the APSDMA and the open weather API. However, due to the dependence on data quality, the application is unable to provide reliable forecasts if there are inaccuracies in the data sourced from APSDMA and the open weather API.

The study [11] presented the development of an Arduino-based weather station based on Internet of Things (IoT) technology for real-time weather monitoring. The system measured and logged critical weather parameters by using sensors consisting of temperature, humidity, atmospheric pressure and rainfall. The data collected by the sensors are transmitted via a Wi-Fi module to the Thing Speak IoT platform that enables remote access to real-time weather information from anywhere in the world. The results of the system evaluation revealed that the integration of IoT with weather monitoring provides a cost-effective solution with various applications, such as in agriculture and disaster management. However, the study is unable to verify the sensor accuracy with a view to expanding the system's capabilities for broader environmental applications.

The study [12] proposed a novel approach to weather forecasting through the use of artificial intelligence, specifically machine learning techniques. The system is developed to be an intelligent weather-predicting module that utilizes statistical measures such as maximum temperature, minimum temperature, and rainfall over a sampled period of days for analysis and prediction. The use of machine learning techniques allows for a prediction accuracy exceeding 90%, which is a significant strength compared to traditional methods. However, users may be unable to achieve such high accuracy if the input data is inadequate or of poor quality. The system's effectiveness is limited by the quality and quantity of the data, making it unable to provide reliable forecasts under suboptimal conditions.

The authors in [13] developed a comprehensive weather monitoring and forecasting system that utilizes Internet of Things (IoT) technology. The system aims to monitor various weather parameters such as temperature, humidity, wind speed, moisture, light intensity, UV radiation, and carbon monoxide levels. It provides real-time data accessible globally through a web interface and a mobile application for alerts on sudden weather changes. The use of IoT allows for continuous monitoring of weather conditions, providing users with up-to-date information that is crucial for decision-making in various sectors such as agriculture, aviation, and disaster management. However, users may be unable to deploy and maintain the sensor devices effectively in remote or harsh environments where technical support is limited. This challenge could hinder the system's ability to provide continuous and accurate weather data in such conditions.

The study [2] developed a web-based application designed to provide accurate weather information tailored to users' locations. The primary goal of the Weather Forecasting application is to deliver precise weather data, including temperature, humidity, wind pressure, wind speed, and times for dawn and sunset, based on the user's geographical location. This is particularly useful for individuals and organizations aiming to mitigate climate-related risks and enhance community benefits. The application utilizes the Open Weather Map API to retrieve weather data specific to the user's location. The data is presented in a user-friendly web interface that is responsive and adapts to various devices, ensuring accessibility for all users. The accuracy and availability of the weather data are dependent on the Open Weather Map API. Any issues with the API, such as downtime or changes in the data structure, could affect the application's performance; while the application provides essential weather parameters, it may not cover all aspects of weather forecasting, potentially limiting its usefulness for users requiring more detailed meteorological data.

The existing related works lack caching mechanism incorporation, which may cause frequent API calls, thereby resulting in higher server costs and ineffective user experience. They also provide related

weather information in the English language only. This makes it difficult for the users without English language understanding.

3. MATERIALS AND METHODS

This chapter delineates the methodological framework employed to develop an improved weather retrieval information application tailored for users in Nigeria.

3.1. Problem Identification

There have been various methodologies employed in weather forecasting applications. One notable study is by [2], which developed a web-based application aimed at providing accurate weather information tailored to users' locations. This application utilizes the Open Weather Map API to retrieve weather data specific to the user's area. The data is presented through a user-friendly web interface that is responsive and adapts to various devices, ensuring accessibility for all users. Additionally, the application includes features that provide specific weather parameters, such as dawn and sunset times, which are particularly useful for users in different geographical settings across the world; the application is also designed as a dynamic site, which means it can change its content and layout in response to user interactions.

Despite this effort, the system fails to provide a caching mechanism that leads to frequent API calls involved, causing higher server costs and ineffective user experience due to slower data retrieval. It is also unable to provide weather information in other languages apart from English language only, which is difficult for those users who do not understand it.

3.2. Proposed System

The proposed system, called an Improved Weather Retrieval Information Application, seeks to address the limitations identified in the existing weather applications developed by [2] by providing an interactive user interface. The system implemented a caching mechanism to store frequently accessed weather data, ensuring faster access and reducing reliance on repeated API calls. This approach enhanced system efficiency, improved localized weather information retrieval across Nigeria, and ensured broader data availability. The caching mechanism will provide support for sectors like agriculture, disaster management, and transportation by providing reliable weather information to aid better decision-making in various activities.

Additionally, the application will provide weather information in major Nigerian languages, such as Hausa, Yoruba and Igbo, to enhance accessibility and inclusivity across diverse user groups.

3.3. System Framework

The proposed system follows a system framework as described and illustrated in Figure 1.

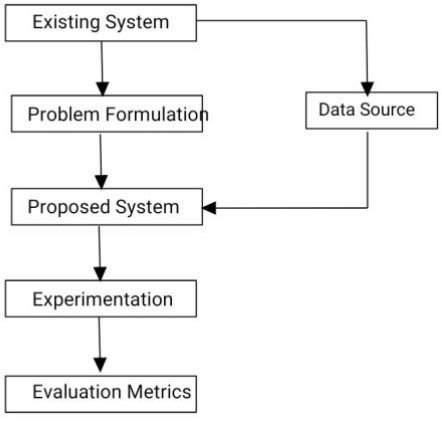

Figure 1. Proposed System Framework

3.4. System Interface Design

The Improved Weather Information Retrieval Application is designed with a focus on accessibility and responsiveness to ensure that users can easily navigate and access essential weather information. The system interface is intuitive, visually appealing, and optimized to be used on various devices, including smartphones, tablets, and desktops. Below is an overview of the key aspects of the system interface design:

3.4.1. User Interface (UI) Layout

The User Interface is designed specifically to ease the means of interaction between the user and the system [14]. The system adopted a clean and minimalist layout to present weather information clearly and concisely to the users. The design follows a card-based structure, where each card displays specific weather details, making it easy for users to scan through information.

3.4.2. Responsive Design

To ensure compatibility across different devices, the system interface employs a responsive design that adjusts its layout based on the device screen size. This ensures that users have a consistent experience whether they are using a smartphone, tablet, or desktop. The application utilizes adaptive scaling and flexible grid layouts, so weather data is always displayed in an organized and readable format, even on smaller screens. The use of touch-friendly controls and gestures also improves navigation on mobile devices, making it easy to access detailed weather forecasts.

3.4.3. Multilingual Support

Multilingual support enables information retrieval systems to allow users to ask and get results in multiple languages [15]. Therefore, the interface is designed to support multiple languages, catering to the diverse linguistic needs of users across Nigeria. Users can easily select their preferred language from the dropdown menu, and the application will automatically adjust the content to display weather information in the chosen language. This multilingual feature ensures that users can interact with the system in Hausa, Yoruba, Igbo, or English, enhancing accessibility and engagement across different communities.

3.5. Application Performance Metrics

Performance metrics focus on the speed and reliability of the weather retrieval application. These metrics measure how quickly the application loads, processes requests, and handles traffic under varying conditions. These include:

Load Time: Measures the time it takes for the weather application to fully load on a user's device.
Response Time: The time it takes for the application to respond to user actions, such as searching for weather data.
Server Response Time: Measures how fast the server processes requests, ensuring minimal delay in delivering weather updates to the user.
Task Completion Rate: The percentage of users who successfully complete a task, such as searching for a city or viewing weather details.
Error Rate: The frequency of errors encountered by users when using the application, such as incorrect city or invalid city.
Time on Task: The time users spend completing a task, which helps evaluate how efficiently the app helps users achieve their goals.
User Satisfaction: User satisfaction is an important metric that reflects how well the application meets the needs and expectations of its users. It can be assessed through user feedback collected through surveys or rating systems that reflect how satisfied users are with the application's performance, interface, and functionality.

3.6. Use Case Diagram

A use case diagram is a visual representation that provides a high-level view of the functionality provided by the system or application from the perspective of its users. It allows developers to describe the possible usage scenarios that a system is developed for" [16]. It illustrates the different ways users can interact with the system and the various scenarios in which the system responds to those interactions. Figure 2 illustrates the use case of the proposed system.

3.7. Data Set

The Improved Weather Retrieval Application leverages the OpenWeatherMap API dataset as its primary data source for delivering real-time and forecasted weather information. This API provides a comprehensive range of meteorological data, such as temperature, humidity, wind speed, air pressure, and precipitation, which are fetched using location-based requests [17]. The system is designed to seamlessly integrate with the OpenWeatherMap API to ensure users receive accurate and localized weather forecasts tailored to various regions across Nigeria.

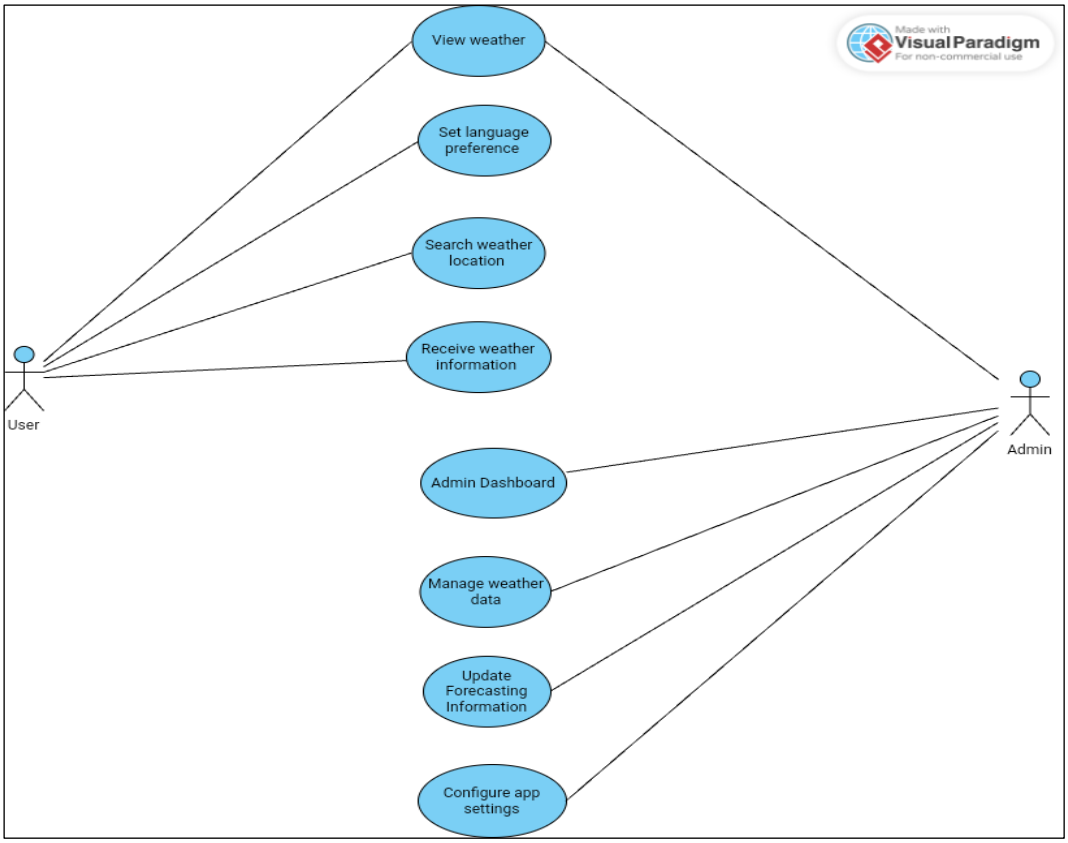

Figure 2. The Use Case Diagram of the Proposed System

3.8. System Interfaces

3.8.1. System Home Page

The Weather Retrieval Information Application's home page serves as the central point of interaction for users. Designed to be user-friendly and intuitive, it allows users to explore weather conditions across various Nigerian cities in multiple languages. Figure 3 presents the proposed system interface known as the home page of the application. Its primary features include:

Weather Search Box: Users can search for weather information using Nigerian cities.
Language Selection: Users can select a preferred language (English, Yoruba, Hausa, or Igbo) to display the weather data.

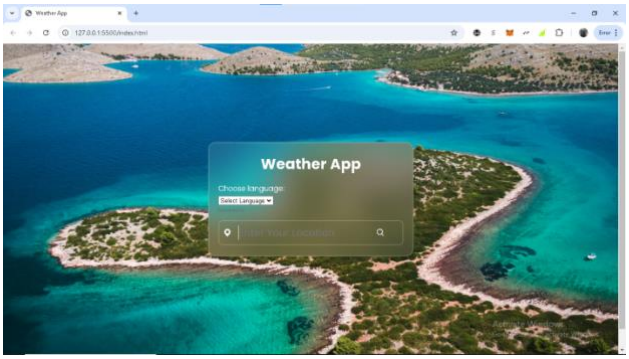

Figure 3. System Home page

3.8.2. Weather Information page

The weather information page serves as the central feature of the application, presenting users with real-time weather data for their inputted locations. It includes dynamic elements that update based on user input and API responses, ensuring an interactive and personalized experience. The page is designed with a

focus on usability, providing intuitive navigation, visually appealing graphics, and concise information to meet users' needs at a glance. Figure 4 presents the Weather Display page of the application.

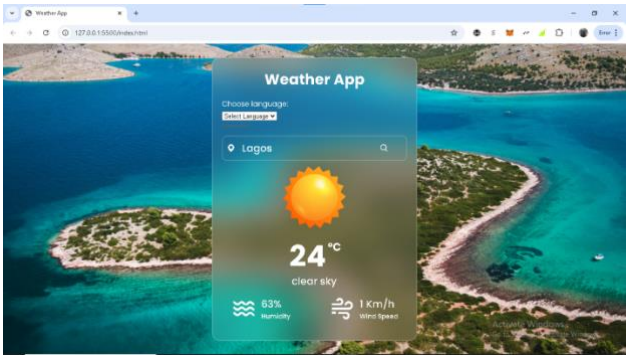

Figure 4. Weather Information page

3.8.3. Error page

The error page is a critical component of the application designed to handle situations where user input or system errors prevent the successful retrieval or display of weather information. It enhances user experience by providing clear and actionable feedback in a visually consistent format. (e.g., a non-existent city or one outside Nigeria). Figure 5 presents the error page of the application while retrieving weather information.

Figure 5. Error page

3.9. Caching Mechanism

The caching mechanism [18] provides efficient weather information retrieval, which is a critical aspect of the application. To minimize redundant API calls and improve the overall performance of the system, a robust caching mechanism is implemented. This ensures that frequently accessed weather data is stored temporarily for quick retrieval.

3.9.1. Integration of Caching Mechanism

The caching mechanism is seamlessly integrated into the application to store weather data retrieved from the OpenWeather API. Each API call result is temporarily saved in the browser's local storage. This integration is designed to reduce the application's dependency on real-time API requests, especially when users frequently search for weather conditions in the same location. The following components are implemented under caching:

- Cache Structure: Weather data is stored as key-value pairs in local storage. This key is a combination of the city name and the value containing the weather data along with a timestamp.
- Data Lifespan: Cached data is assigned a lifespan of 30 minutes. After this duration, the cache expires, and a fresh API call is triggered.
- Automatic Expiry Check: The system checks the timestamp associated with cached data before using it. If the data is older than the defined lifespan, it is discarded and replaced with new data.
- Fallback to Real-Time API: In case of a cache miss (i.e., if no valid data is found in the cache), the system directly fetches the data from the weather API and caches it for future use.

3.9.2. Functionality of the Caching Mechanism

- Reduce API Call Frequency: By storing recently fetched weather data, the mechanism avoids redundant API requests, significantly reducing network latency and improving application performance.
- Improve User Experience: Cached data ensures that users experience faster response times, especially when revisiting a previously searched city or refreshing the page.
- Enhance Reliability: In cases of temporary API unavailability or slow network connectivity, cached data serves as a fallback, ensuring users still receive weather information.

3.10. Translation Mechanism

The machine-translation mechanism provides translation for the document by using a machine-translation system [19]. In this study, the translation mechanism in the weather information retrieval application leverages Google Translate to dynamically convert the interface text and weather conditions into supported languages such as English, Hausa, Igbo, and Yoruba. This ensures accessibility for users from different linguistic backgrounds in Nigeria.

The translation mechanism integrates Google Translate API to handle dynamic translations in real time. Users can select their preferred language from a dropdown menu, which triggers the translation of both static and dynamic text elements on the interface.

3.10.1. English Implementation

English serves as the default language for the application. Google Translate is not triggered when English is selected, as the system is already designed with English as the base language. Weather descriptions fetched from the API are displayed directly in English.

3.10.2. Hausa Translation Implementation

Hausa translation enables users in Northern Nigeria to interact with the system in their native language. When Hausa is selected, Google Translate converts all interface texts and weather conditions to Hausa. This ensures users who speak Hausa can navigate and understand the information effortlessly. Figure 6 presents the Hausa Language Translation Implementation interface.

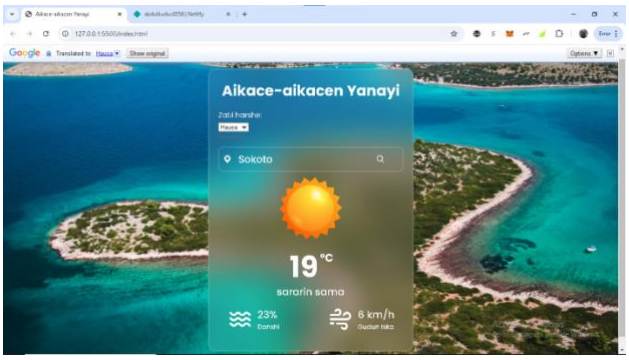

Figure 6. Hausa Translation Implementation Interface

3.10.3. Igbo Translation Implementation

For users in Southeastern Nigeria, the system supports Igbo translations. Selecting Igbo activates the Google Translate API to dynamically translate the system's static and dynamic content into Igbo. This includes labels, buttons, and real-time weather descriptions, ensuring clarity and usability. Figure 7 presents the Igbo Language Translation implementation interface.

3.10.4. Yoruba Translation Implementation

The Yoruba translation is tailored for users in Southwestern Nigeria. Selecting Yoruba activates Google Translate to convert all visible text and weather-related information into Yoruba. This allows Yoruba users to interact with the system in their native language seamlessly. Figure 8 presents the Yoruba Language Translation Implementation interface.

Figure 7. Igbo Translation Implementation Interface

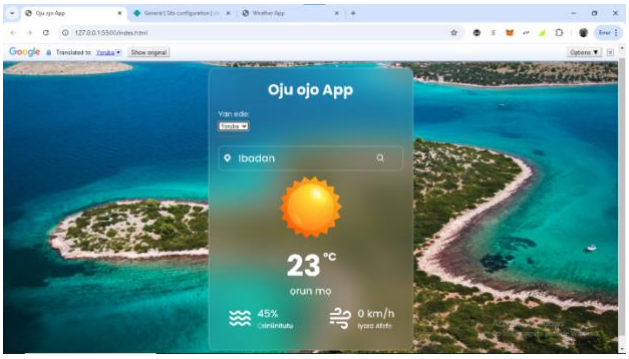

Figure 8. Yoruba Translation Implementation Interface

4. RESULTS AND DISCUSSION

System testing is a software testing phase that assesses the functionality of an integrated software system. The aim of this testing is to determine whether the system adheres to the stated requirements. The implementation of this system involves two types of testing.

4.1. Unit Testing

Unit testing is used to validate the correctness of basic units or components of the developed system under test [20]. It ensures that each feature or functionality performs as intended, independently of the overall system. In this application, unit testing was performed on key functionalities like the weather display, translation mechanism, error handling, and caching system. Table 1 presents the proposed system unit testing result.

Table 1. Table showing unit test results

Test	Expected Results	Remark
Weather Data Fetch	The system should retrieve accurate weather data from the API for a given Nigerian city, the weather information, including temperature, description, humidity, and wind.	Successfully
User Input Validation	The system should validate user input to ensure it matches a valid Nigerian city.	Successfully
Language Dropdown	The user should be able to select a language from the dropdown, and the system updates according to the selected language.	Successfully
Translation Mechanism	The interface should translate content dynamically into English, Hausa, Igbo, or Yoruba based on user selection.	Successfully
Error Handling for Invalid Location	The application should display an error message for invalid or non-Nigerian cities entered by the user.	Successfully

Caching Mechanism	Weather data should be retrieved from local storage if available and not older than 30 minutes.	Successfully
Weather Icon Display	The correct weather icon should be displayed based on the weather condition (e.g. clear sky, rain).	Successfully

4.2. Performance Testing

Performance testing serves as a critical phase in evaluating and ensuring the efficiency and reliability of applications under diverse conditions. For this study, performance testing was conducted on two systems: the existing system, which fails to address key aims and objectives, and the proposed weather information retrieval system, which integrates features such as translation and caching mechanisms tailored to Nigeria's unique needs. Using WebPageTest as the primary tool, the performance of both systems was analyzed through metrics such as page load times, resource optimization, and user experience under various conditions like network speeds and geographic locations. This comparative testing approach highlighted the strengths and weaknesses of both systems in real-world scenarios. The performance test was carried out for both the existing system and the newly developed system.

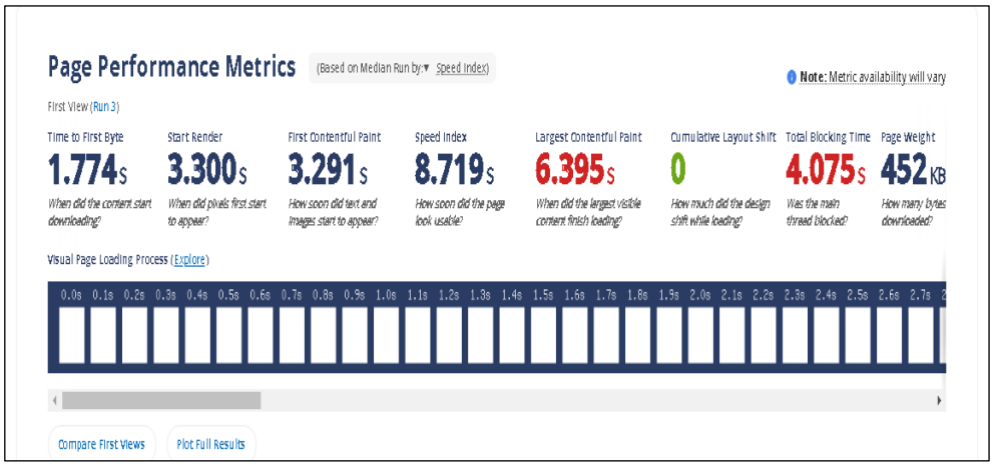

Figure 9. Page Performance Metrics for the existing system

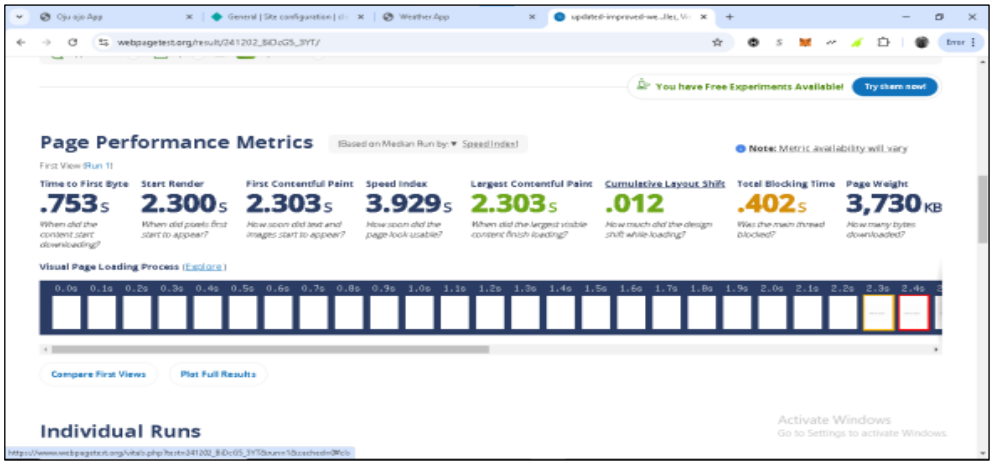

Figure 10. Page Performance Metrics for the proposed system

Figure 9 and Figure 10 compared the load times, resource optimization, and overall responsiveness of both the existing and the improved weather retrieval system. The existing system exhibits slower load times and higher resource consumption due to its lack of caching mechanisms and optimization strategies. These inefficiencies become more apparent under low-bandwidth conditions, negatively impacting user experience. In contrast, the improved system demonstrates significantly reduced load times and efficient resource utilization, achieved through the integration of caching mechanisms. This ensures that weather data is quickly retrieved without the need for repeated API calls while optimizing the performance of the system.

Figure 11. Connection view for the Existing system

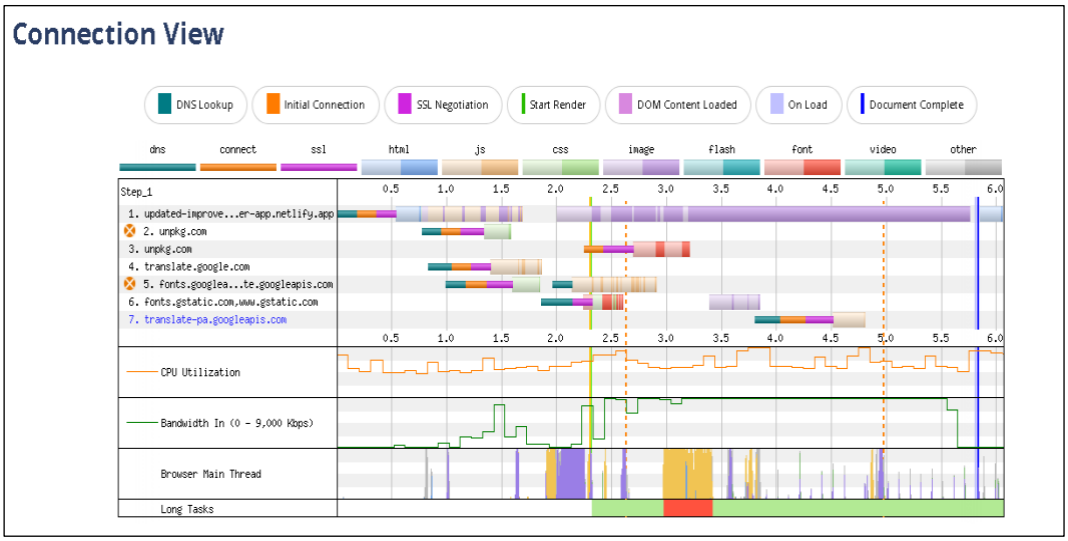

Figure 12. Connection view for the proposed system

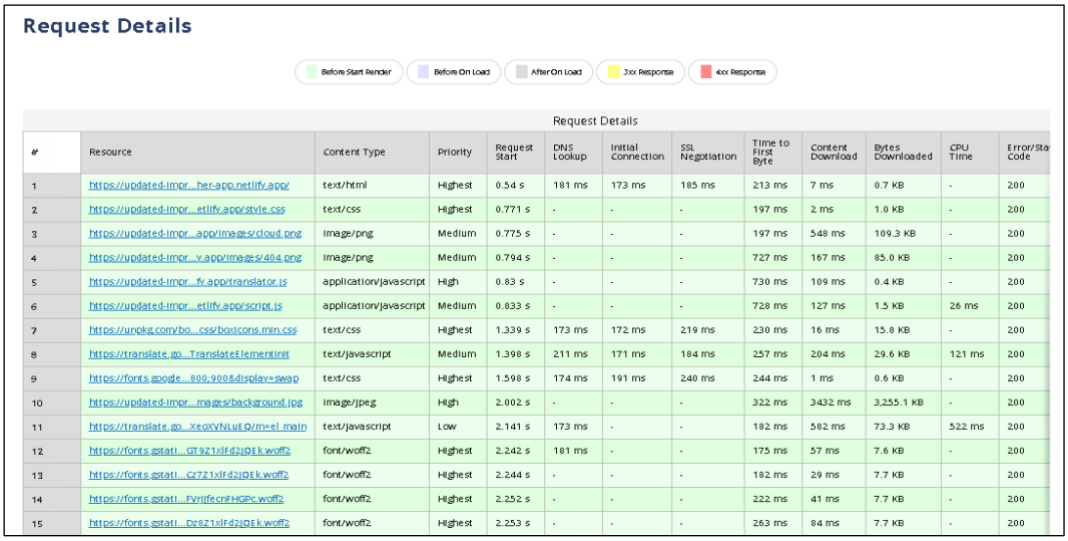

Figure 13. Request Details for the proposed system

Figure 11 and Figure 12 provided a visual breakdown of network requests made by both systems during performance testing. For the existing system, the diagram reveals frequent and redundant API calls that increase latency and server load. This inefficiency is particularly problematic in low-resource settings. On the other hand, the improved system utilizes caching to minimize redundant network requests, leading to faster responses and reduced server dependency. The streamlined request patterns in the new system ensure a smoother experience for users, especially those in rural areas in Nigeria.

Figure 13 highlights the types and number of requests made during the operation system. The existing system's reliance on live API calls for every weather update results in unnecessary overhead and delays. In contrast, the improved system stores frequently accessed data locally through caching, reducing the number of external requests required. This approach not only improves speed but also enhances reliability by providing users with cached weather data during network disruptions. Additionally, the new system incorporates language translations, ensuring inclusivity and making weather information accessible to Nigeria's diverse population.

5. CONCLUSION

This study represented a comprehensive effort to develop a weather information retrieval application that meets the linguistic and geographical needs of Nigerian users. By integrating real-time weather data retrieval, caching mechanisms, and multilingual support, the application ensures a seamless user experience across diverse demographics. The application GUI, coupled with performance optimization through the caching mechanism and WebPageTest tool, has been a cornerstone of its success. In essence, the system showcases the potential of leveraging technology to bridge accessibility gaps and provide localized solutions to weather forecasting challenges. This underscores the commitment to developing an inclusive, efficient, and user-centric application. The results of the evaluation revealed that the existing system took a time of 1.774s at the speed of 8.719s to retrieve relevant weather information, while the proposed system took a lesser time of 0.753s at a faster speed of 3.929s to retrieve relevant weather information. Results demonstrated the proposed system's ability to deliver accurate weather information with faster data retrieval and improved usability compared to existing systems. It also effectively addressed the challenges of multilingual accessibility and user interface complexity, providing a reliable and user-friendly platform. However, further research in this direction can enhance its effectiveness by expanding language options to include additional regional dialects, which will promote inclusivity. Integrating advanced features such as severe weather alerts, predictive analytics, and extended forecast options can also be explored.

DATA AVAILABILITY STATEMENT

The data used in this study are available at OpenWeatherMap API.

CONFLICTS OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

[1] O. O.A., O. I.O., and F. O.A., "Review of Climate Change and Its effect on Nigeria Ecosystem," *Int J Rural Dev Environ Heal Res*, vol. 3, no. 3, pp. 92–100, 2019, doi: 10.22161/ijreh.3.3.3.

[2] Priyanka Gharad, Suchita Bawne, and Poorvi Meshram, "Designing a Weather Forecasting Application using API," *Int J Adv Res Sci Commun Technol*, pp. 205–210, Apr. 2024, doi: 10.48175/IJARSCT-17234.

[3] S. Panchalwar, "Weather App Using Android Studio," *Gurukul Int Multidiscip Res J*, Jun. 2024, doi: 10.69758/GIMRJ2406I8V12P090.

[4] Sunil Khatri, "Advanced Weather Forecasting with Machine Learning: Leveraging Meteorological Data for Improved Predictions," *Commun Appl Nonlinear Anal*, vol. 32, no. 5s, pp. 01-16, Dec. 2024, doi: 10.52783/cana.v32.2912.

[5] M. S. Murthy, R. P. R. Kumar, B. Saikiran, I. Nagaraj, and T. Annavarapu, "Real Time Weather Monitoring System using IoT," *E3S Web Conf*, vol. 391, p. 01142, Jun. 2023, doi: 10.1051/e3sconf/202339101142.

[6] S. Meshram, "Weather Forecasting in Python Django with Machine Learning," *Int J Res Appl Sci Eng Technol*, vol. 10, no. 6, pp. 487–489, Jun. 2022, doi: 10.22214/ijraset.2022.43749.

[7] M. S. S, "Weather Forecast and it's Visualization using Augmented Reality: Mobile App," *INTERANTIONAL J Sci Res Eng Manag*, vol. 08, no. 03, pp. 1–11, Mar. 2024, doi: 10.55041/IJSREM29086.

[8] G. K. Rahul, S. Singh, and S. Dubey, "Weather Forecasting Using Artificial Neural Networks," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, IEEE, Jun. 2020, pp. 21–26. doi: 10.1109/ICRITO48877.2020.9197993.

[9] S. S. Vallabh S., Shreenidhi R., Tejasvi U, , Kaustubh P., Swayam P., Shraddha Y., "Weather Displaying Platform," *Int J Creat Res Thoughts*, vol. 12, no. 5, pp. 1–3, 2024.

[10] N. Kumar, M. Chodisetti, T. Kumarsai, and V. Garg, "Data-Driven Approaches to Weather Prediction With Machine Learning," in *2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, IEEE, Feb. 2025, pp. 575–580. doi:

10.1109/CICTN64563.2025.10932643.
[11] Silvia Ganesan, Chong Peng Lean, L. Chen, Kong Feng Yuan, Ng Poh Kiat, and Mohammed Reyasudin Basir Khan, "IoT-enabled Smart Weather Stations: Innovations, Challenges, and Future Directions," *Malaysian J Sci Adv Technol*, pp. 180–190, Apr. 2024, doi: 10.56532/mjsat.v4i2.293.
[12] T. R. V. Anandharajan, G. A. Hariharan, K. K. Vignajeth, R. Jijendiran, and Kushmita, "Weather Monitoring Using Artificial Intelligence," in *2016 2nd International Conference on Computational Intelligence and Networks (CINE)*, IEEE, Jan. 2016, pp. 106–111. doi: 10.1109/CINE.2016.26.
[13] Balakrishnan Sivakumar and Chikkamadaiah Nanjundaswamy, "Weather monitoring and forecasting system using IoT," *Glob J Eng Technol Adv*, vol. 8, no. 2, pp. 008-016, Aug. 2021, doi: 10.30574/gjeta.2021.8.2.0109.
[14] A. Almu and K. Maiyama, "A Textual Case-Based Mobile Phone Diagnosis Support System," *Niger J Basic Appl Sci*, vol. 18, no. 2, Mar. 2011, doi: 10.4314/njbas.v18i2.64336.
[15] J.-Y. Nie and F. Jin, "A Multilingual Approach to Multilingual Information Retrieval," 2003, pp. 101–110. doi: 10.1007/978-3-540-45237-9_8.
[16] M. Seidl, M. Scholz, C. Huemer, and G. Kappel, "The Use Case Diagram," 2015, pp. 23–47. doi: 10.1007/978-3-319-12742-2_3.
[17] U. Sharma, E. Das, Diksha, and P. Yadav, "INTERACTIVE WEATHER FORECASTING SYSTEM USING OPENWEATHER API AND WEB TECHNOLOGIES," *Int J Res -GRANTHAALAYAH*, vol. 13, no. 1, Jan. 2024, doi: 10.29121/granthaalayah.v13.i1.2025.6119.
[18] N. Dutta, H. K. D. Sarma, R. Jadeja, K. Delvadia, and G. Ghinea, "Caching Mechanisms for Faster Content Retrieval," 2021, pp. 95–118. doi: 10.1007/978-3-030-46736-4_5.
[19] M. Madankar, M. B. Chandak, and N. Chavhan, "Information Retrieval System and Machine Translation: A Review," *Procedia Comput Sci*, vol. 78, pp. 845–850, 2016, doi: 10.1016/j.procs.2016.02.071.
[20] H. Yu *et al.*, "Automated assertion generation via information retrieval and its integration with deep learning," in *Proceedings of the 44th International Conference on Software Engineering*, New York, NY, USA: ACM, May 2022, pp. 163–174. doi: 10.1145/3510003.3510149.

BIOGRAPHIES OF AUTHORS

Abba Almu received his PhD in Computer Science from Usmanu Danfodiyo University, Sokoto. Masters Degree in Computing: Information Engineering from Robert Gordon University, Aberdeen, United Kingdom. He also received his B.Sc. (Hons) in Computer Science from Usmanu Danfodiyo University, Sokoto. He is currently a Reader at the Department of Computer Science, Faculty of Physical and Computing Sciences of Usmanu Danfodiyo University, Sokoto, Nigeria. His research interests include information retrieval, recommender systems and machine learning. He can be contacted at email: almu.abba@udusoku.edu.ng.

Abdulkudus Yunusa Received a B.Sc. Computer Science from the Department of Computer Science of Usmanu Danfodiyo University, Sokoto. His research interests include web development and programming. He can be contacted at email: justqudus@yahoo.com.