IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 12, December 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.524

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312154|

Weather Application Using Full Stack Development

Shruthi M K, Rohini A, Basantha Kumari, Beena Sheril, Dr. Suresh G B MCA., Ph.D.

Associate Professor, Department of Computer Science, S.J.M.I.T, Chitradurga, Karnataka, India

M. Tech, 3rd Semester, Department of Computer Science, SJMIT, Chitradurga, Karnataka, India

Associate Professor, Department of Computer Science, S.J.M.I.T, Chitradurga, Karnataka, India

Assistant Professor, Department of Computer Science, S.J.M.I.T, Chitradurga, Karnataka, India

Professor, College of Computing & Information Science, University of Technology & Applied Sciences, Ibra, Sultanate of Oman

ABSTRACT: This weather application is designed to provide real-time weather information on a cross-platform environment, supporting Windows XP, Windows, and Mac operating systems. Developed using HTML, CSS, and JavaScript, the application ensures a responsive and user-friendly experience. The integrated development environment (IDE) options include Visual Studio Code, Sublime Text, Atom, and JetBrains, allowing flexibility for developers to choose their preferred environment.

The application leverages the Open Weather Map API to fetch current weather data, ensuring accurate and up-to-date information. Users can easily obtain weather details such as temperature, humidity, wind speed, and forecasts for a specified location. The Google Chrome browser is recommended for optimal performance.

By combining a modern tech stack and an intuitive user interface, this weather application aims to deliver a seamless experience for users across different operating systems, fostering accessibility and convenience. Whether users are on Windows XP, Windows, or Mac, they can rely on this application to stay informed about the latest weather conditions. The use of widely accepted development tools and the React library ensures maintainability and extensibility, making it a versatile solution for developers interested in weather-related applications.

KEYWORDS: Visual Crossing Weather API, Current temperature, humidity, wind speed.

I. INTRODUCTION

The Weather Application is a user-friendly tool designed to provide quick and accurate real-time weather information for any location. Its intuitive interface allows users to easily input their desired location or enable automatic detection. Upon launch, essential weather metrics such as temperature, humidity, wind speed, and precipitation chances are displayed prominently, offering a comprehensive snapshot of current conditions.

For those seeking more detailed forecasts, the app provides hourly and daily breakdowns, including temperature trends, precipitation probabilities, and wind patterns. Powered by the reliable Visual Crossing Weather API, the Weather Application ensures users receive up-to-date and accurate weather updates, allowing them to stay informed and prepared for any weather conditions.

In a world where weather conditions play a significant role in our daily lives, having quick and accessible information about the current weather is crucial. The Weather Application is designed to meet this need, offering users a simple and efficient way to check real-time weather details for a specific location. Built with a user-friendly interface and powered by a reliable weather API, this application provides up-to-date information, ensuring users are well-informed and prepared for the day's weather conditions.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312154|

The Weather Application's intuitive design allows users to effortlessly input their desired location or enable location services for automatic detection. Upon launching, users are greeted with a clean interface showcasing essential weather metrics at a glance. Current temperature, humidity, wind speed, and precipitation chances are prominently displayed, providing a comprehensive snapshot of the immediate conditions.

For those seeking a more detailed forecast, the app offers hourly and daily breakdowns. Users can explore temperature trends, precipitation probabilities, and wind patterns over the coming hours and days, aiding in planning activities or making informed decisions about travel and outdoor engagements.

The Weather Application's reliability stems from its integration with a trusted and robust weather API that is, Visual Crossing weather API. This partnership guarantees accurate and real-time updates, ensuring users receive the most current weather information available.

In general, the Weather Application stands as an indispensable tool for anyone needing instant and reliable weather updates ensuring that users stay ahead of severe weather conditions, allowing them to take necessary precautions promptly.

II. LITERATURE SURVEY

In a world where weather conditions play a role in our daily lives, having quick and accessible information about the current weather is crucial. The Weather Application is designed to meet this need, offering users a simple and efficient way to check real-time weather details for a specific location. Built with a user-friendly interface and powered by a reliable weather API, this application provides up-to-date information, ensuring users are well-informed and prepared for the day's weather conditions.

The Weather Application's intuitive design allows users to effortlessly input their desired location or enable location services for automatic detection. Upon launching, users are greeted with a clean interface showcasing essential weather metrics at a glance. Current temperature, humidity, wind speed, and precipitation chances are prominently displayed, providing a comprehensive snapshot of the immediate conditions.

For those seeking a more detailed forecast, the app offers hourly and daily breakdowns. Users can explore temperature trends, precipitation probabilities, and wind patterns over the coming hours and days, aiding in planning activities or making informed decisions about travel and outdoor engagements.

The Weather Application's reliability stems from its integration with a trusted and robust weather API that is, Visual Crossing Weather API. This partnership guarantees accurate and real-time updates, ensuring users receive the most current weather information available.

In general, the Weather Application stands as an indispensable tool for anyone needing instant and reliable weather updates ensuring that users stay ahead of severe weather conditions, allowing them to take necessary precautions promptly.

III. METHODOLOGIES

This document details the development of a user-centric weather application, designed to bridge the gap between users and real-time weather information. The application prioritizes a set of core objectives to deliver a seamless and valuable user experience.

Prioritizing User Needs:

- **Accessibility and User-Friendly Interaction:** At the forefront lies the goal of making weather information accessible to everyone. A simple and intuitive interface, coupled with effortless city input and clear data presentation, minimizes user effort and ensures anyone can grasp weather conditions quickly.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312154|

Reliable and Up-to-Date Data:

- **Real-Time Data Retrieval and Error Handling:** Integration with a reputable weather API, like Visual Crossing, guarantees users receive the most current weather data available. Robust error handling mechanisms gracefully manage situations where API requests fail, providing informative messages to guide users.

Technical Enhancements for a Flawless Experience:

- **Responsiveness and Security:** The application's design ensures a visually-appealing and seamless experience across various devices and screen sizes. Security measures safeguard sensitive information, such as API keys, to prevent unauthorized access and maintain application integrity.
- **Enhanced Development with React:** The application leverages the React library to bolster its performance, modularity, and maintainability. This approach facilitates future improvements and simplifies the development process.By achieving these objectives, this weather application empowers users with readily available and dependable weather information. This empowers informed decision-making for daily activities, allowing users to plan with confidence.

- **Accessibility:**

Make weather information easily accessible to users by providing a simple and intuitive interface.

- **User-Friendly Interaction:**

Create a user-friendly experience, allowing users to input a city name effortlessly and receive relevant weather details with minimal effort.

- **Real-Time Data Retrieval:**

Utilize a weather API (such as Visual Crossing Weather API) to fetch up-to-date weather data, ensuring that users receive the most current and accurate information available.

- **Data Presentation:**

Present weather data in a clear and understandable format, including essential information such as temperature, weather description.

- **Responsiveness:**

Design the application to be responsive, ensuring a seamless and visually appealing experience across various devices and screen sizes.

- **Input Validation:**

Implement input validation to ensure that users provide a valid city name before initiating the weather data request, enhancing the reliability of the application.

- **Error Handling:**

Include robust error handling mechanisms to gracefully manage scenarios where the API request fails or encounters issues. Provide informative error messages to guide users in such situations.

- **Security:**

Ensure the secure handling of sensitive information, such as API keys, to prevent unauthorized access and maintain the integrity of the application.

- **Enhancement:**

Employ the React library to enhance the application's performance, modularity, and maintainability.

IV. CHOSEN SYSTEM DESIGN

➢ **SYSTEM OVERVIEW:**

- **User Interface:** Developed using HTML for structure and Tailwind CSS for styling, providing an intuitive and visually appealing interface.
- **Frontend Logic:** Implemented using JavaScript and React, facilitating interactive user experiences and dynamic content rendering.
- **Data Fetching:** Utilizes Axios to asynchronously fetch weather data from the Visual Crossing Weather API.
- **External Service Integration:** Integrates with the Visual Crossing Weather API to retrieve accurate weather data.
- **Deployment Environment:** Deployed on web servers accessible via Google Chrome, with development facilitated through Visual Studio Code.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312154|

➢ **COMPONENT DETAILS:**

- **User Interface (UI):**
  - Developed using HTML for structural markup, organizing elements such as buttons, forms, and containers.
  - Styled using Tailwind CSS, leveraging utility classes for rapid and consistent styling across components.
- **Frontend Logic (JavaScript and React):**
  - Implemented using React, a JavaScript library for building user interfaces.
  - Components such as App.jsx and WeatherCard.jsx manage data fetching and rendering.
  - Utilizes React hooks (e.g., useState, useEffect) for managing state and side effects.
- **Data Fetching (Axios):**
  - Axios, a promise-based HTTP client, is used to asynchronously fetch weather data from the Visual Crossing Weather API.
  - Implements error handling and response parsing to ensure robust data retrieval.
- **External Service Integration (Visual Crossing Weather API):**
  - Integrates with the Visual Crossing Weather API to obtain weather data for specified locations.
  - Requires an API key for authentication and access to API endpoints.
- **Deployment Environment:**
  - Developed and tested for deployment on web servers accessible via Google Chrome.
  - Utilizes Visual Studio Code as the primary development environment, providing features such as syntax highlighting, code completion, and integrated terminal support.

➢ **INTEGRATION WITH VISUAL CROSSING WEATHER API:**

- The Weather Application integrates with the Visual Crossing Weather API to fetch weather data.
- Requires registration with Visual Crossing Weather API to obtain an API key for authentication.
- Implements API requests to specific endpoints (e.g., current conditions, forecasts) based on user input and application requirements.
- Handles API responses, parsing JSON data for display within the application's UI.

➢ **DEVELOPMENT ENVIRONMENT:**

- Visual Studio Code serves as the primary development environment for coding, debugging, and version control.

Google Chrome is used for testing and deployment, ensuring compatibility and performance across web browsers.

V. DATA FLOW DIAGRAM

DETAILED DESCRIPTION OF COMPONENTS / SUBSYSTEMS

➢ **Weather application components Description:**

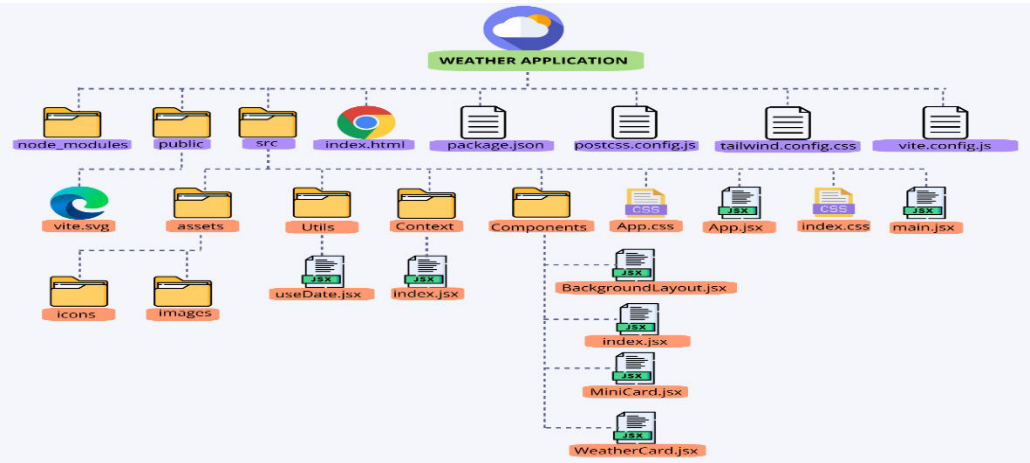


Fig 2.1 Weather application Components Diagram

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

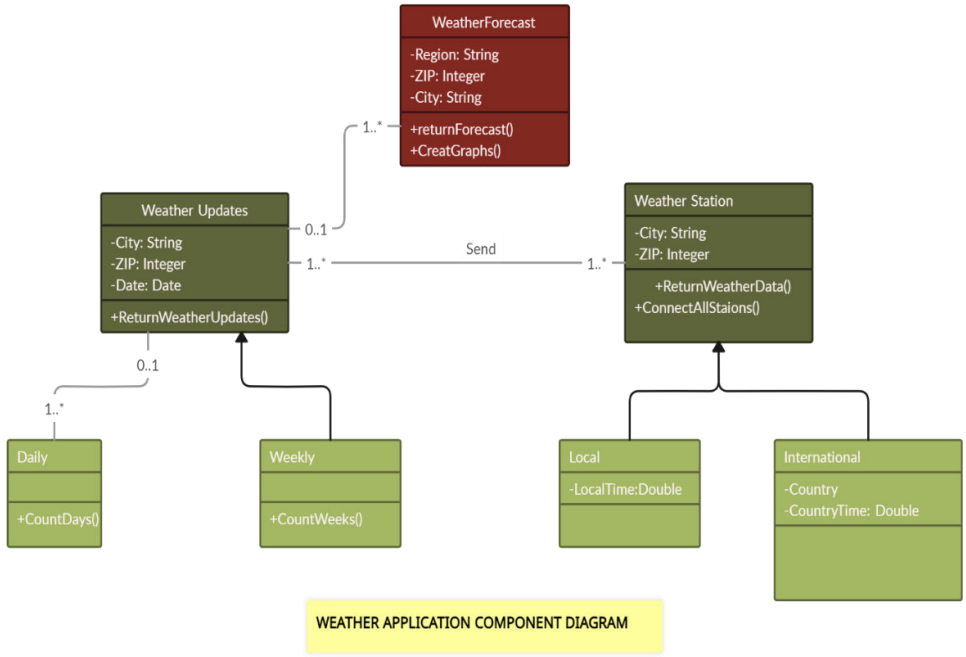
|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312154|

- **Node modules:** This folder contains the third-party libraries that are used by the application.
- **Public:** This folder contains the files that are accessible to the user, such as the HTML file and the CSS files.
- **package.json:** This file contains information about the application, such as its name, version, and dependencies.
- **Src:** This folder contains the source code for the application, such as the JavaScript files and the JSX files.
  - **Assets:** This folder contains the images and icons that are used by the application.
- **Icons:** This folder contains the icons that are used by the application
- **Images:** Thisfolder contains the images that are used by the application
  - **Utils:** This folder contains utility functions that are used throughout the application.
- **useDate.jsx:** This file contains a custom hook that is used to format the date.
  - **Context:** This folder contains the context API, which is used to share data between different parts of the application.
- **index.jsx:** This file is the entry point of the application.
  - **Components:** This folder contains the reusable components that make up the user interface of the application.
- **BackgroundLayout.jsx:** This file contains the component that renders the background of the application.
- **MiniCard.jsx:** This file contains the component that renders a mini weather card.
- **WeatherCard.jsx:** This file contains the component that renders a weather card.
- **index.jsx:** This file is the entry point of the application.
  - **App.css:** This file contains the CSS styles for the application.
  - **App.jsx:** This file contains the main component of the application
  - **index.css:** This file contains the CSS styles for the index component.
  - **main.jsx:** This file contains the main JavaScript code for the application.

COMPONENT 1- N



WEATHER APPLICATION COMPONENT DIAGRAM

- **Weather Application:** This component manages the overall weather application and interacts with other components to retrieve and display weather data. It provides methods like returnForecast() to retrieve weather forecasts and createGraphs() to generate visual representations of the weather data.
- **Weather Updates:** This component aggregates weather data from various sources, including Daily, Weekly, Local, and International components. It provides methods like returnWeatherUpdates() to access the aggregated weather data.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312154|

- **Weather Station:** This component represents individual weather stations that collect raw weather data like temperature, humidity, and pressure. It provides a method returnWeatherData() to share the collected data with other components.
- **Daily, Weekly, Local, and International:** These components represent different types of weather forecasts. They likely have methods related to calculating and storing weather data specific to their timeframes and locations.

VI. PROPOSED SYSTEM

1. Current weather:
- City name or user location
- Temperature (Celsius or Fahrenheit)
- Weather description (e.g., sunny, cloudy, rainy)
- Humidity
- Wind speed and direction
- Icon representing weather conditions

2. Hourly forecast:
- Temperature for the next 24 hours
- Chance of precipitation
- Weather icons for each hour

3. Daily forecast:
- High and low temperatures for the next 5-7 days
- Chance of precipitation each day
- Weather icons for each day

4. User interface:
- Clean and well-organized layout
- Easy navigation between screens
- Visually appealing design

5. Additional weather data:
- UV index
- Air quality
- Pollen count
- Visibility
- Sunrise and sunset times

6.Multiple location support:
- Ability to track weather in multiple cities or regions

7.Charts and graphs:
- Visualize trends in weather data over time

ADVANTAGES OF THE PROPOSED SYSTEM

Cross-Platform Compatibility:

- By utilizing a Full Stack Development approach, you ensure that your weather application can be seamlessly deployed and accessed across various operating systems, including Windows XP, Windows, and Mac.

Flexibility with Text Editors and IDEs:

- FSD allows developers to work with a range of popular text editors and Integrated Development Environments (IDEs) such as Visual Studio Code, Sublime Text, Atom, and Jetbrains. This flexibility in tooling makes it easier for developers to choose the environment that suits their preferences and workflow.

Efficient API Integration:

- Full Stack Development enables efficient integration of external APIs, such as the OpenWeatherMap API Key. This ensures real-time and accurate weather data retrieval for the application, enhancing its reliability and usefulness.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.524

|www.ijirset.com |A Monthly, Peer Reviewed & Referred Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 13, Issue 12, December 2024

|DOI: 10.15680/IJIRSET.2024.1312154|

Seamless Collaboration with Version Control:

- Full Stack Development promotes the use of version control systems like Git, allowing developers to collaborate seamlessly. This ensures that the development process is organized, changes are tracked, and multiple developers can work on different aspects of the application simultaneously.

Optimized Code with Visual Studio Code and PyCharm:

- The use of Visual Studio Code and PyCharm as Integrated Development Environments enhances code quality and efficiency. These IDEs provide advanced features such as syntax highlighting, code completion, and debugging tools, contributing to optimized and well-maintained code.

Browser Compatibility

- FSD ensures that the weather application is compatible with popular browsers, such as Google Chrome. This compatibility extends the reach of the application to a wider audience, ensuring a consistent user experience regardless of the browser.

Enhanced User Experience:

- With a dynamic and responsive user interface, real-time weather updates, and efficient data retrieval through API integration and ReactJS implementation in the weather application developed using FSD provides an enhanced user experience, making it more appealing and user-friendly.

Rapid Development and Deployment:

- Full Stack Development streamlines the development process, enabling rapid prototyping and deployment. This agility is crucial for staying ahead of competitors and adapting to changing user needs and market demands.

VII. CONCLUSION

The development of this weather application successfully delivers a user-centric tool for accessing real-time weather information. The application prioritizes a simple and intuitive interface, effortless data retrieval, and clear presentation, ensuring a smooth user experience for anyone seeking current weather conditions.

Integration with a reliable weather API guarantees up-to-date forecasts, while robust error handling mechanisms provide informative messages in case of unforeseen issues. Furthermore, the application's responsive design ensures a seamless experience across various devices.

By leveraging the React library, the application boasts enhanced performance, modularity, and maintainability, paving the way for future improvements and a user-friendly weather experience.

REFERENCES

1. https://gemini.google.com/app
2. https://chat.openai.com/
3. https://rapidapi.com/visual-crossing-corporation-visual-crossing-corporation-default/api/visual-crossing-weather/
4. https://www.geeksforgeeks.org/forecast-weather-project-check-today-weather-for-any-location/
5. https://dev.to/imshines/a-simple-weather-app-using-react-and-openweathermap-api-10m2
6. Christina Mary Jolly, Safna K.M, Dr. S. Brilly Sangeetha, Weather Prediction and Climate Analysis Using Machine Learning.
7. Gaurav Kumar BHARTI,1, Abhijeet RANJAN, Anshul BHARAT, Suraj YADAV, Design and Development of an Efficient and Intelligent Weather Forecasting App 2023.
8. Tulsi Pawan Fowdur, Rosun Mohammad Nassir-Ud-Diin Ibn Nazir, A real-time collaborative machine learning based weather forecasting system with multiple predictor locations, July 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details