

Sinhgad Technical Education's

Smt. Kashibai Navale Collage of Engineering Pune.



DEPARTMENT OF COMPUTER ENGINEERING

Mini Project Report

On

“Weather Application”

310257: Web Technology Laboratory

Submitted By

Bhaiyasaheb Chaudhari C34423

Prathmesh Swami C34424

Prathmesh Thikekar C34429

Kalpesh Padmawar C34432

**Under Guidance Of
Prof. Priyanka Jadhav**

Prof. Priyanka Jadhav

Prof. R. H. Borhade

Subject Teacher

HOD, Computer

Table of Contents

Sr. No	content	Page No.
1	Introduction	
	1.1 Problem Statement	1
	1.2 Proposed System	1
	1.3 Advantages	
	1.4 System Architecture	2
2	Hardware and Software	
	2.1 Hardware Requirements	3
	2.2 Software Requirements	3
3	Overall Description	
	3.1 Description	4
4	Outcome	
	4.1 Screen shots	6
	4.2 Code	
5	Conclusion	9
6	References	10



CERTIFICATE

This is to certify that the project report entitles

Weather Application

submitted by	
Bhaiyasaheb Chaudhari	C34423
Prathmesh Swami	C34424
Prathmesh Thikekar	C34429
Kalpesh Padmawar	C34432

is a bonafide work carried out by above student under the supervision of Prof.

Priyanka S. Jadhav and it is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, Pune for the award of the degree of Bachelor of Engineering (Computer Engineering).

Prof. Priyanka S. Jadhav
Project Guide

Prof. R.H.Borhade
Head of Department

Place: Pune

Date: /04/2023

Abstract

The weather app is a web application that provides users with real-time weather updates for any location in the world. The app uses APIs to collect weather data from multiple sources and presents it in a simple and user-friendly format. Users can input a location and receive current and forecasted weather information, such as temperature, humidity, wind speed, and precipitation. The purpose of this project is to design and develop a weather app that is responsive, reliable, and intuitive to use. The app will be built using modern web development technologies, including HTML, CSS, and JavaScript, and will leverage various APIs to gather and present weather data. The end product will be a web application that is accessible to anyone with an internet connection and a desire to know the weather forecast for their location.

A weather app is a type of application that provides users with real-time weather information for a specific location. These apps typically rely on weather data from various sources, including national weather services and private weather companies, and use APIs to collect and present this information in an easily digestible format.

Keywords:

API (Application Programming Interface) , API_key, API call etc.

1. Introduction

1.1 Problem Statement

The web application will be designed with a user-centric approach, offering an intuitive interface that allows users to easily access and understand weather data. It will be optimized for both mobile and desktop devices, ensuring that users can access the web application from anywhere and at any time. Additionally, the web application will be equipped with advanced features such as push notifications for severe weather alerts and a customizable widget for quick access to weather information. The ultimate goal of the project is to provide users with a comprehensive and reliable weather web application that enhances their daily lives by keeping them informed and prepared for any weather condition.

1.2 Proposed System

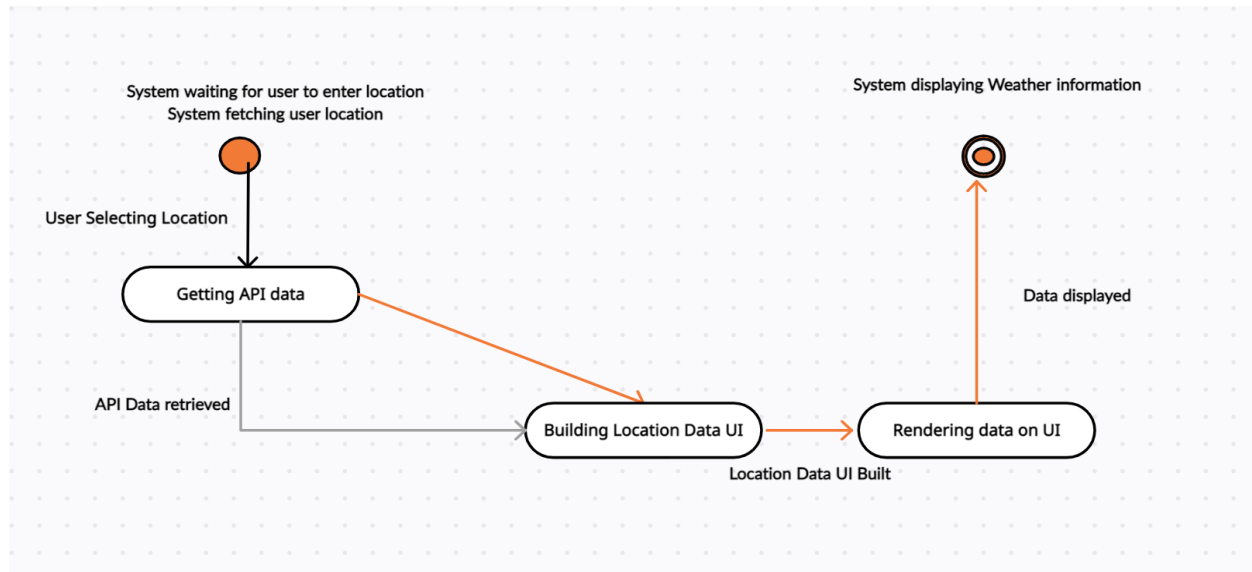
The web application will also incorporate advanced features such as personalized weather details, interactive weather maps, and detailed daily forecasts. The personalized weather details will allow users to save their favorite locations and receive weather updates for those locations. The interactive weather maps will provide users with a visual representation of weather patterns in their area, and the detailed daily forecasts will offer users a more comprehensive overview of the weather conditions for the day.

To ensure that the web app is reliable and up-to-date, it will make API calls to trusted weather data sources such as OpenWeather API. The web app will also be optimized for both mobile and desktop devices to provide users with a seamless experience regardless of their device. Overall, the proposed system aims to provide users with a comprehensive and user-friendly weather web application that enhances their daily lives by keeping them informed and prepared for any weather condition.

1.3 Advantages

1. **Reliable and Accurate:** The web application leverages trusted weather data sources such as OpenWeather API, ensuring that users receive accurate and up-to-date weather information.
2. **User-Friendly:** The web application is designed with a user-centric approach, providing an intuitive interface that allows users to easily access and understand weather data.
3. **Advanced Features:** The web application includes advanced features such as personalized weather details, interactive UI, and detailed daily forecasts, making it stand out from other weather apps and providing a better user experience.
4. **Cross-Platform Compatibility:** The web application is optimized for both mobile and desktop devices, ensuring that users can access it from anywhere and at any time.
5. **Customizable:** With the inclusion of advanced features such as push notifications for severe weather alerts and customizable widgets, the web application offers a high level of convenience and customization to its users.

1.4 The system architecture is shown in the diagram below:



- The user interacts with the web application via the user interface, which sends requests to the OpenWeather API directly from the client-side JavaScript. The API call retrieves the weather data, which is then parsed and displayed in the user's browser using HTML and CSS.
- The proposed system does not include advanced features such as interactive maps or personalized alerts, as these are not part of your project. However, these features could be added in future iterations of the project if desired.
- Overall, the proposed system for your weather app web application is designed to provide users with a reliable, easy-to-use, and feature-rich weather app that leverages data from trusted weather data sources such as OpenWeather API to ensure accurate and up-to-date weather information.

2. Hardware and Software Requirements

2.1 Hardware Requirements

Hardware Requirements	Specification
Computer	Intel Pentium 4 or later processor
RAM	Minimum 1 GB or higher
Display	Minimum 1024 x 768 resolution
Internet Connection	Broadband or High-Speed Connection

2.2 Software Requirements

Software Requirements	Specification
Operating System	Windows 7 or higher, Mac OS X, Linux
Web Browser	Google Chrome, Mozilla Firefox, Safari, Microsoft Edge
Code Editor	Visual Studio Code, Sublime Text, Atom
Libraries/Frameworks	OpenWeather API
Programming Languages	HTML, CSS, JavaScript

3. Overall Description

3.1 Description

Module 1 - switchTab:

This module is responsible for switching between two user interfaces. The first user interface is the auto location taking UI, which is displayed when the application is launched. The second user interface is the weather details UI, which is displayed when the weather information is fetched from the server. The switchTab module listens for user clicks on the tabs and toggles the visibility of the UIs accordingly.

Module 2 - fetchUserWeatherInfo:

This module is responsible for fetching the weather information from the server. It gets the location of the user either through geolocation or user input, and then makes API calls to the OpenWeather API to retrieve the weather information in JSON format. The module then returns the JSON object to the calling entity.

Module 3 - renderWeatherInfo:

This module is responsible for rendering the weather information on the UI. It takes the JSON object returned by the fetchUserWeatherInfo module and extracts the relevant information such as the temperature, weather condition, and wind speed. The module then updates the UI with this information using HTML and CSS.

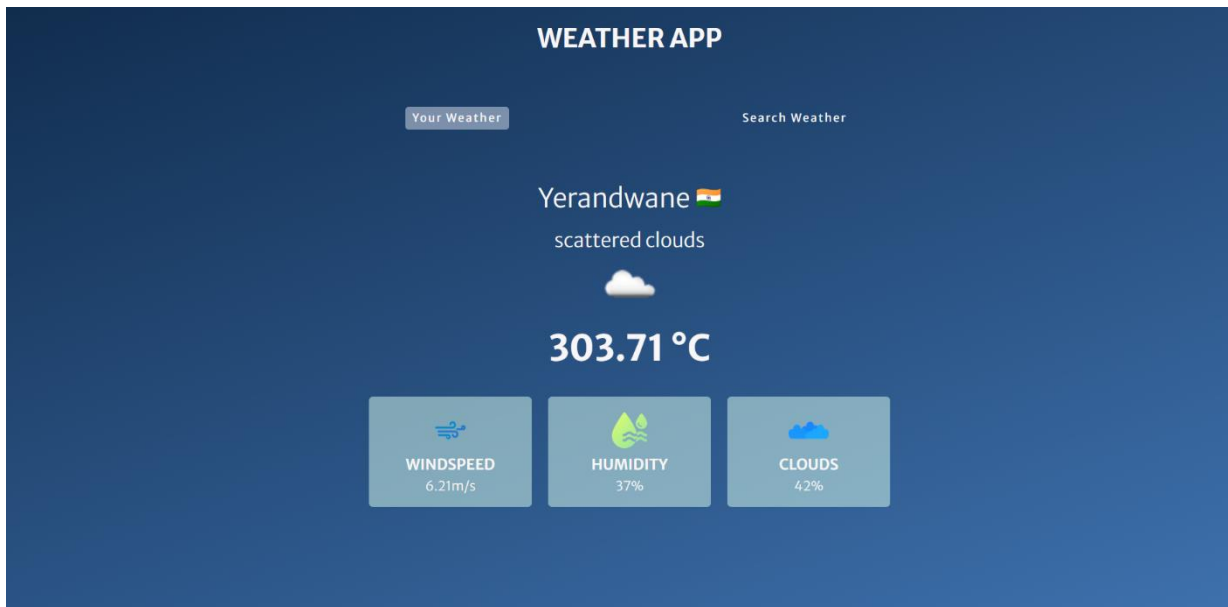
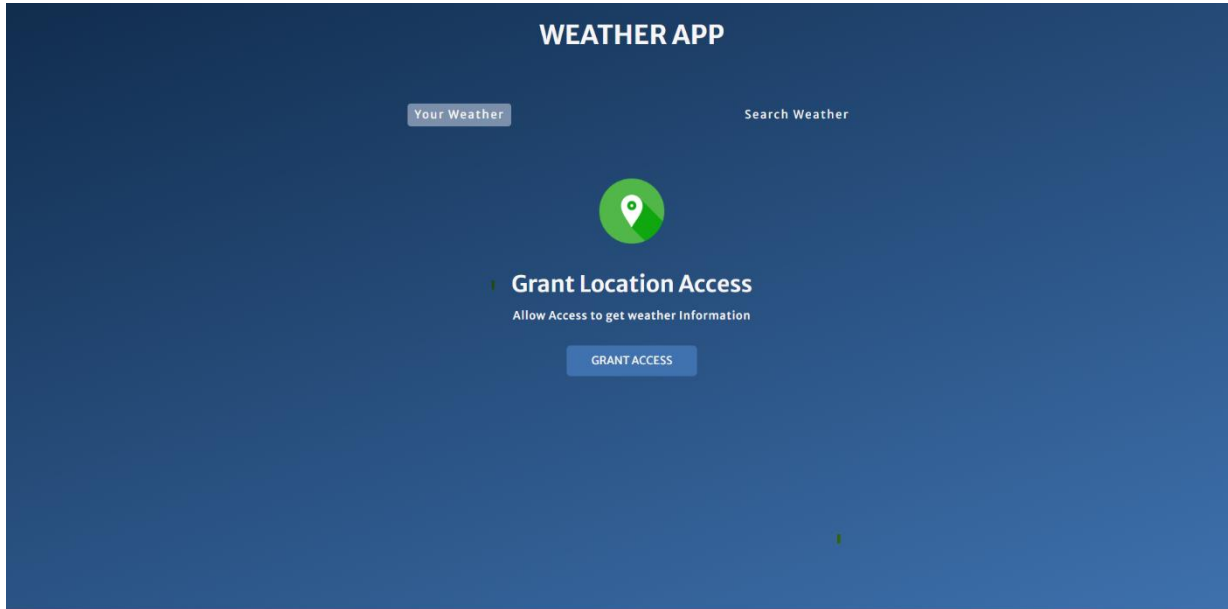
Module 4 - getLocation:

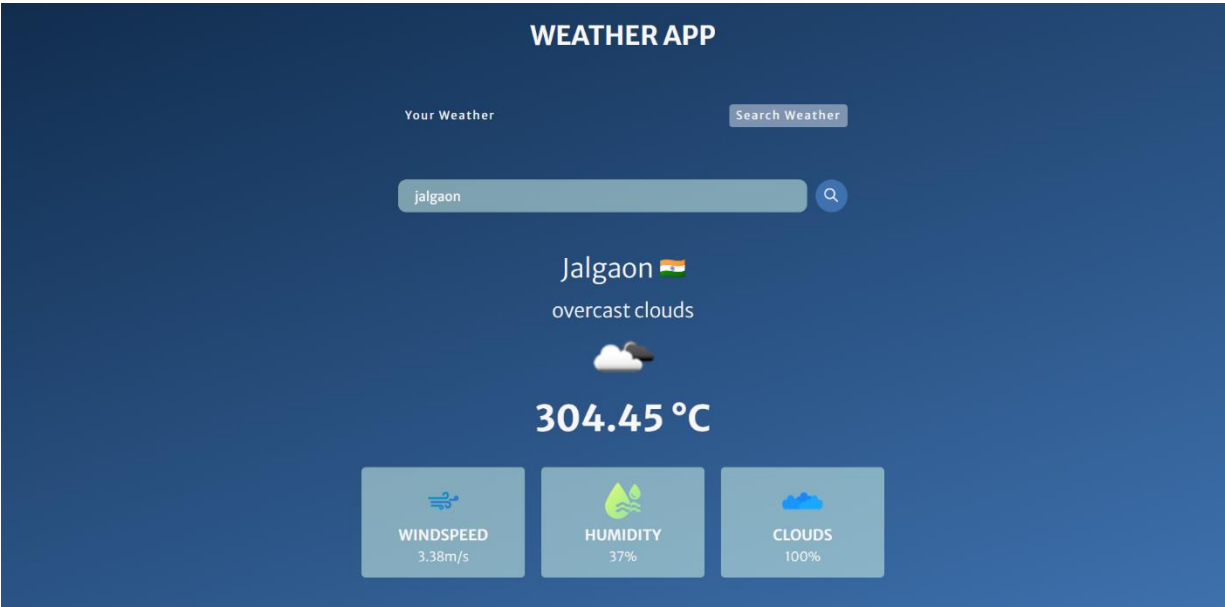
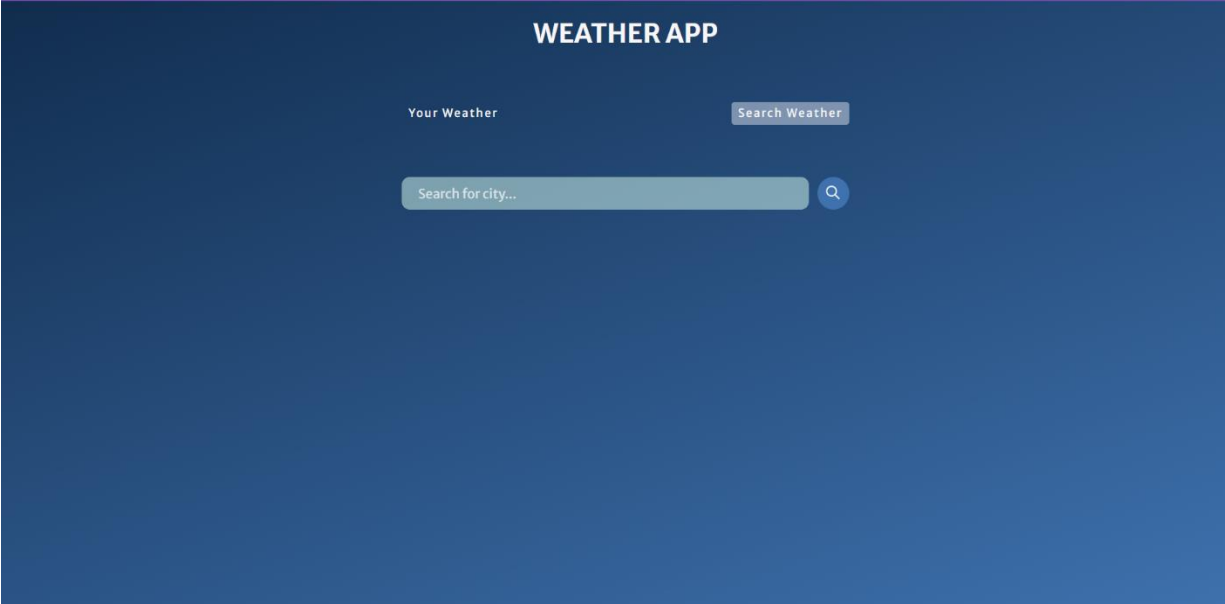
This module is responsible for finding the user's location using geolocation. It first checks if the user's browser supports geolocation, and then requests permission to access the user's

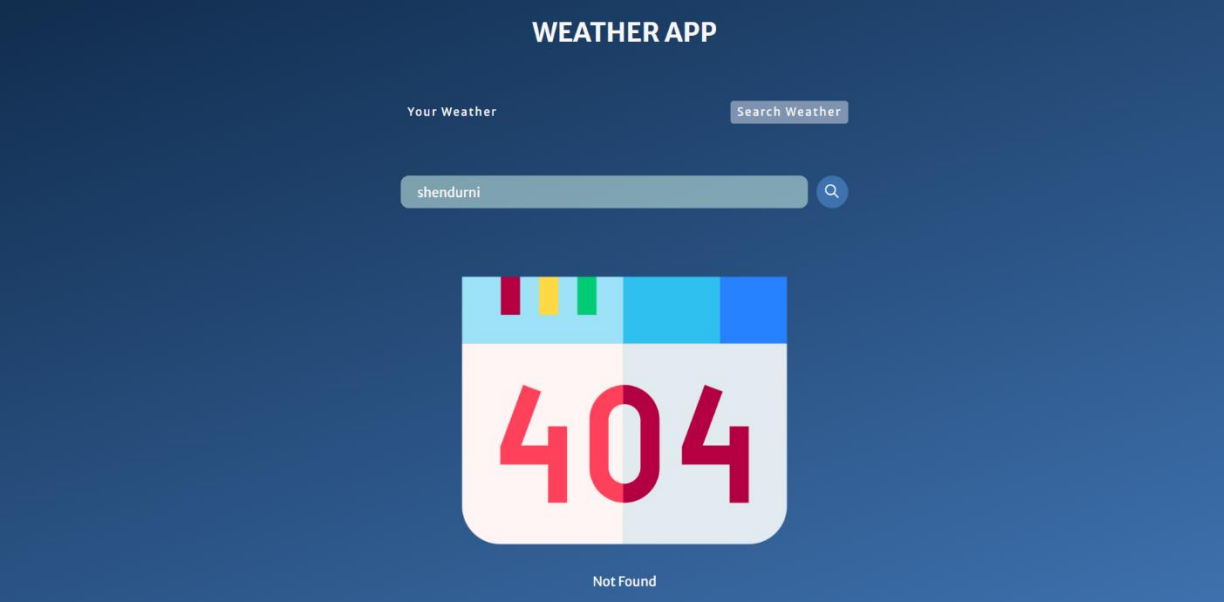
location. If permission is granted, the module retrieves the latitude and longitude of the user's location and provides it to the calling entity.

4. Outcome

4.1 Screen shots







5. Conclusion

In conclusion, the proposed weather web application aims to provide users with a comprehensive and user-friendly platform that enhances their daily lives by keeping them informed and prepared for any weather condition. By leveraging data from trusted weather data sources such as OpenWeather API, the web application will ensure accurate and up-to-date weather information. The advanced features such as personalized weather details, interactive UI, and detailed daily forecasts will make the web application stand out from other weather apps and provide a better user experience.

Overall, the weather web application is designed with a user-centric approach, providing an intuitive interface that allows users to easily access and understand weather data. The web application will be optimized for both mobile and desktop devices, ensuring that users can access it from anywhere and at any time. With the inclusion of advanced features such as push notifications for severe weather alerts and customizable widgets, the web application will offer a high level of convenience and customization to its users.

6. References.

1. "Building Web Applications with Visual Studio 2017: Using .NET Core and Modern JavaScript Frameworks" by Philip Japikse, Kevin Grossnicklaus, and Ben Dewey - This book includes a chapter on building a weather web application using .NET Core and Angular.
2. "JavaScript: The Definitive Guide: Activate Your Web Pages" by David Flanagan - This book includes a section on using JavaScript to access weather data from web services and display it on web pages.
3. "HTML5 and CSS3 Responsive Web Design Cookbook" by Benjamin LaGrone - This book includes a chapter on building a responsive weather web application using HTML, CSS, and JavaScript.
4. OpenWeather API - OpenWeather API provides weather data for developers to use in web applications. It offers both free and paid plans, and supports a variety of programming languages including JavaScript, PHP, and Python.