

**A  
REPORT  
ON  
Weather App**

*Submitted in the partial fulfillment of requirement for the award of the  
degree of*

**Bachelor of Technology (B.Tech.)**

**in**

**Computer Science and Engineering**



**Submitted By:**

**Hariom Singh (2115000428)**

**Submitted To:**

**Mr. Amir Khan  
Dept. of CEA**

## DECLARATION OF STUDENT

I hereby declare that the work, which is being presented in the Project Report, entitled "**Weather App**" in partial fulfilment for the award of Degree of Bachelor of Technology" in Dept. Of Computer Science and Engineering and submitted to the Department of Computer Science and Engineering, GLA University, is a record of my own investigations carried under the Guidance of **Mr. Amir Khan** Department of Computer Science and Engineering,

### **GLA University**

We have not submitted the matter presented in this Project Report anywhere for the award of any other Degree.

**(Name and Signature of Candidates)**

**Hariom Singh (2115000428)**

**Counter Signed by**

**Mr. Amir Khan**

# PREFACE

The Weather App project is a web application that leverages HTML, CSS, and JavaScript to fetch weather data from a weather API and display current weather conditions and forecasts for a specific location. It provides users with real-time weather information, allowing them to stay informed about the weather conditions in their desired location.

The Weather App utilizes an API (in this example, the Weather API) to retrieve weather data based on the user's input location. The application dynamically fetches the current weather information and the forecast for the upcoming days. It then presents this data in a user-friendly format, making it easy for users to understand and interpret.

**Chapter 1** provides the details regarding the introduction of the area of project, Software tools and technology used like React and HTML, CSS, JS.

**Chapter 2** provides problem statement, Objective in term of functional and non-functional requirements provided by the project

**Chapter 3** includes various UML diagrams like use case diagrams, activity diagram, description about tables, ER diagrams and flow charts.

**Chapter 4** includes testing, where the various modules testing is done to check whether the is according to the user, Unit testing and Integration testing is done on modules using different test cases.

**Chapter 5** includes results with proper screen shots and description about the screenshots like its purpose, how this screen will appear etc.

**Chapter 6** clearly defines the steps required to make settings in the pc to run your project along with the snapshot. In short, anybody can install and run our project without our support by just referencing this chapter.

## ACKNOWLEDGEMENT

Engineers in all disciplines must acquire knowledge of project making. Student, in particular, will find ‘project making’ as an integral part of their studies that will infuse the spirit of doing practical work in them.

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible whose constant guidance crowned our efforts with success.

We sincerely express our deep gratitude to the management of our college for giving us liberty to choose and to work on the most relevant project i.e. **“Weather App”**. We are thankful to **Mr. Rohit Agarwal (HOD CSE)** for ensuring that we have a smooth environment at the college and lab. At the very outset we would like to offer our never ending thanks to our project supervisor **Mr. Amir Khan** (Assistant Professor, CSE) who helped us with our project from the beginning till the end. His continuous surveillance over our work allowed us to work more efficiently.

**Hariom Singh**

**Sec - H**

## **Bonafide Certificate**

This certificate is to certify that **Hariom Singh** is a bonafide student of **GLA University** pursuing the **BTech** program. **Hariom Singh** has been enrolled in the program since **14-9-2021** and is currently in the **5th/3 Year** semester.

**Hariom Singh** has a good academic record and is actively involved in extracurricular activities. [He/She] is a well-mannered and respectful student who is always willing to help others.

This certificate is issued at the request of **Hariom Singh** for the purpose of **Project Report**.

**Supervisor's Signature:**

**SUPERVISOR**

Mr. Amir Khan

Computer Science and Engineering

GLA University

Date:06-12-2023

## ABSTRACT

Weather prediction is a critical component of our daily lives, influencing decision-making across various sectors. This comprehensive review explores the advancements in weather prediction methodologies, technologies, and their diverse applications. The scope spans from short-term forecasts for immediate planning to long-term climate predictions, considering the integration of cutting-edge technologies and evolving scientific understanding.

The review delves into the principles of meteorology, emphasizing the importance of understanding atmospheric phenomena. It examines the evolution of numerical weather prediction (NWP) models, highlighting the role of mathematical simulations and data assimilation techniques in enhancing prediction accuracy.

Technological advancements, including supercomputing capabilities, satellite technologies, and sophisticated sensors, are integral to the progress in weather prediction. Remote sensing instruments contribute real-time data, enriching the input for predictive models.

## LIST OF FIGURES

| <b>Fig. No.</b> | <b>Figure Description</b> | <b>Page No.</b> |
|-----------------|---------------------------|-----------------|
| Figure 1.1      | System Diagram            | 2               |
| Figure 1.2      | JSON Object Diagram       | 4               |
| Figure 2.2      | UI Design                 | 12              |

# TABLE OF CONTENTS

|  |     |
|--|-----|
| Certificate .....                                  | i   |
| Declaration of student .....                       | ii  |
| Preface.....                                       | iii |
| Acknowledgement.....                               | iv  |
| Abstract.....                                      | v   |
| List of Figures.....                               | vi  |
| 1. Introduction.....                               | 1   |
| 1.1 Introduction.....                              | 1   |
| 1.2 Key Features.....                              | 1   |
| 1.3 Benefits .....                                 | 1   |
| 1.4 Technologies Used .....                        | 3   |
| 1.3.1 HTML and CSS.....                            | 3   |
| 1.3.2 Javascript.....                              | 6   |
| 1.3.3 Json.....                                    | 7   |
| 1.3.3 Weather API.....                             | 8   |
| 2. Problem Identification.....                     | 9   |
| 2.1 Problem Statement.....                         | 9   |
| 2.2 Project Scope.....                             | 9   |
| 2.3 Design and Implementation Constraint.....      | 10  |
| 3. Design.....                                     | 11  |
| 3.1.1Proposed approach.....                        | 11  |
| 3.1.2 Weather Prediction System Architecture ..... | 12  |
| 3.3 UI Design .....                                | 13  |
| 4. Testing. ....                                   | 14  |
| 4.1 Unit Testing.....                              | 14  |
| 4.2Integration Testing .....                       | 14  |
| 5. Results .....                                   | 15  |
| 6. Deployment.....                                 | 18  |
| 6.1 Purpose .....                                  | 18  |
| 6.2 Preparation and Procedure .....                | 18  |
| 6.3 Environment Variables.....                     | 18  |
| 7 Conclusion .....                                 | 20  |
| 8 Future Scope.....                                | 21  |
| 9 Reference and Github Links .....                 | 22  |
| 10 Appendix... ..                                  | 23  |



# Chapter 1

## Introduction

---

The Weather App project is a web application that leverages HTML, CSS, and JavaScript to fetch weather data from a weather API and display current weather conditions and forecasts for a specific location. It provides users with real-time weather information, allowing them to stay informed about the weather conditions in their desired location.

The Weather App utilizes an API (in this example, the Weather API) to retrieve weather data based on the user's input location. The application dynamically fetches the current weather information and the forecast for the upcoming days. It then presents this data in a user-friendly format, making it easy for users to understand and interpret.

### Key Features:

1. **Location Input:** Users can enter the desired location for which they want to fetch weather information. The app validates the input and prompts users to provide a valid location if necessary.
2. **Current Weather Display:** The application fetches and displays the current weather conditions for the specified location. It provides essential information such as temperature, weather condition, and location details.
3. **API Integration:** The app integrates with a weather API (e.g., WeatherAPI) to retrieve weather data. It sends requests to the API, receives the responses containing weather information, and parses the data for display.
4. **Responsive Design:** The Weather App is designed to be responsive, ensuring that it works well on different devices and screen sizes. Users can access the app and view weather information on their desktops, laptops, tablets, or mobile devices.

### Benefits:

- **Real-time Weather Information:** Users can obtain up-to-date weather data for any desired location, enabling them to make informed decisions based on current and forecasted weather conditions.
- **User-friendly Interface:** The application offers a clean and intuitive interface for entering locations and viewing weather information. It presents data in a clear and organized manner, enhancing the user experience.
- **Planning and Preparedness:** The Weather App allows users to plan their activities and make informed decisions based on weather forecasts. They can adapt their schedules, clothing, or travel plans accordingly, enhancing safety and convenience.

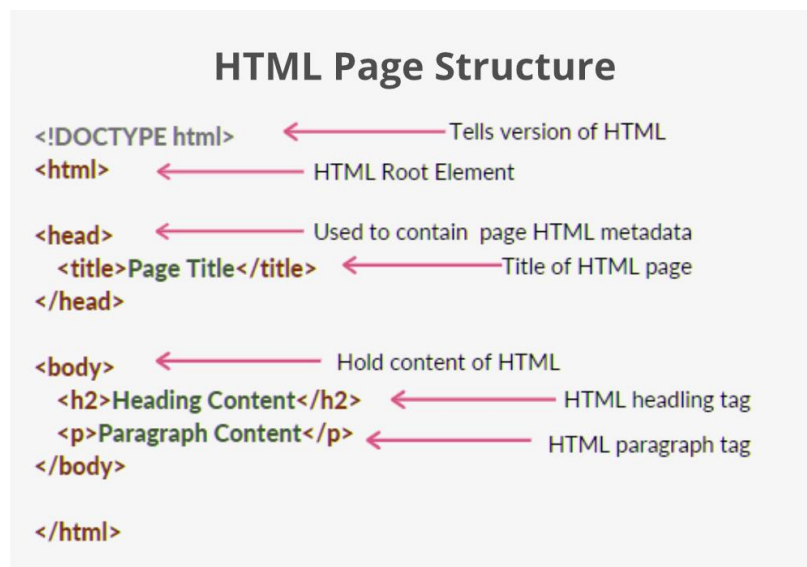
- API Integration: By integrating with a weather API, the app can leverage the data and capabilities provided by the API, ensuring accurate and reliable weather information.

The Weather App project offers a practical and useful tool for users who want to stay informed about weather conditions. It demonstrates the power of HTML, CSS, and JavaScript in building web applications that interact with APIs and provide real-time data to enhance user experiences.

## Technologies Used

### 1.3.1 HTML

HTML, which stands for HyperText Markup Language, is the standard markup language used to create and design documents on the World Wide Web. It provides a standardized way to structure content on the web, allowing browsers to interpret and display the content consistently. HTML is a fundamental technology that works in conjunction with other web technologies like CSS (Cascading Style Sheets) and JavaScript to build interactive and visually appealing web pages.



**Markup Language:** HTML is a markup language that uses a system of tags to define the structure of content on a web page. Tags are enclosed in angle brackets, and they instruct the browser on how to render and display different elements.

**Document Structure:** HTML documents have a hierarchical structure that consists of various elements such as headings, paragraphs, lists, images, links, forms, and more. These elements help organize and present content on a web page.

**Tags:** HTML tags are used to define elements in a document. Tags usually come in pairs: an opening tag and a closing tag. The content is placed between these tags.

**Attributes:** HTML tags can include attributes that provide additional information about an element. Attributes are placed within the opening tag and usually come in name/value pairs

**Web Page Skeleton:** Every HTML document typically starts with a document type declaration (`<!DOCTYPE html>`) and contains the `<html>`, `<head>`, and `<body>` elements. The `<head>` section includes metadata, while the `<body>` section contains the visible content.

**Versioning:** HTML has evolved over the years, and different versions have been released. The latest version is HTML5, which introduces new features and enhancements, including improved support for multimedia and interactive content.

**Cross-Browser Compatibility:** HTML is designed to be interpreted consistently across different web browsers, ensuring that web pages look and behave similarly regardless of the browser being used.

**Compatibility with CSS and JavaScript:** HTML works in conjunction with CSS for styling and layout and JavaScript for interactivity. The combination of HTML, CSS, and JavaScript forms the core technologies for building modern, dynamic web pages.

## CSS

CSS, or Cascading Style Sheets, is a style sheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG or XHTML). CSS defines how elements should be displayed on a webpage, including layout, colors, fonts, spacing, and other visual aspects. By separating the content (HTML) from its presentation (CSS), web developers can create more flexible and maintainable websites.

Here are key concepts and features of CSS:

**Selectors:** CSS uses selectors to target HTML elements for styling. Selectors can be based on element names, classes, IDs, attributes, and more

**Properties and Values:** CSS rules consist of properties and values. Properties define the style aspects (e.g., color, font-size), and values specify the settings for those properties. Multiple property-value pairs are enclosed within curly braces.

**Box Model:** The CSS box model describes how elements are structured on a webpage. It includes properties like margin, border, padding, and width/height. Understanding the box model is crucial for designing layouts.

**Layout:** CSS enables the creation of flexible and responsive layouts. Techniques like Flexbox and Grid provide powerful tools for building complex and responsive designs.

**Responsive Design:** With media queries, CSS allows developers to apply different styles based on the characteristics of the device, such as screen size, resolution, or orientation. This facilitates the creation of responsive and mobile-friendly websites.

**Transitions and Animations:** CSS supports transitions and animations to create dynamic and interactive elements on a webpage. This includes effects like fade-ins, slide-outs, and more.

**Selectors and Specificity:** The specificity of a selector determines which styles apply when multiple rules conflict. Understanding specificity is crucial to resolving styling conflicts.

**Vendor Prefixes:** Some CSS features require vendor prefixes to work consistently across different browsers. For example, `-webkit-` for WebKit-based browsers like Chrome and Safari.

**CSS Preprocessors:** CSS preprocessors like Sass or Less extend the capabilities of traditional CSS, offering features like variables, mixins, and nested rules to enhance the maintainability and organization of stylesheets.

**CSS Frameworks:** Frameworks like Bootstrap and Foundation provide pre-written, customizable styles and components, helping developers build consistent and visually appealing websites more efficiently.

### 1.3.2 Javascript

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform. Once you learnt JavaScript, it helps you developing great front-end as well as back-end software using different JavaScript based frameworks like jQuery, Node.JS etc.

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation
- Displaying pop-up windows and dialog boxes

JavaScript is a versatile and widely-used programming language that is primarily known for its role in web development. It is a scripting language that allows developers to add dynamic content, interactivity, and client-side functionality to websites. Here are some key aspects of JavaScript:

**Client-Side Scripting:** JavaScript is mainly used as a client-side scripting language, meaning it runs in the user's web browser. It enables dynamic interactions and modifications to the content of a webpage without the need for server-side processing.

**Object-Oriented Language:** JavaScript is an object-oriented language, which means it is based on the concept of objects. Objects encapsulate data and behavior, providing a structured way to organize and manipulate code.

**Syntax and Structure:** JavaScript has a C-style syntax with curly braces {}, and it follows a procedural, functional, and object-oriented programming paradigm. It is designed to be lightweight and flexible.

**Variables and Data Types:** Variables are used to store data in JavaScript. The language supports various data types, including numbers, strings, booleans, objects, arrays, and more.

**Functions:** Functions in JavaScript allow developers to group code into reusable blocks. Functions can be declared and called to perform specific tasks.

**Event Handling:** JavaScript is commonly used for handling events on web pages, such as button clicks, mouse movements, and keyboard inputs. Event handlers are functions that are triggered in response to specific events.

**DOM Manipulation:** The Document Object Model (DOM) represents the structure of an HTML document. JavaScript allows developers to manipulate the DOM dynamically, enabling the creation, modification, and deletion of HTML elements.

**Asynchronous Programming:** JavaScript supports asynchronous programming through features like callbacks, promises, and async/await. This is crucial for handling tasks such as fetching data from a server without blocking the execution of other code.

**Frameworks and Libraries:** There are many JavaScript frameworks and libraries, such as React, Angular, and Vue.js, that simplify and enhance the development of complex web applications by providing reusable components and efficient state management.

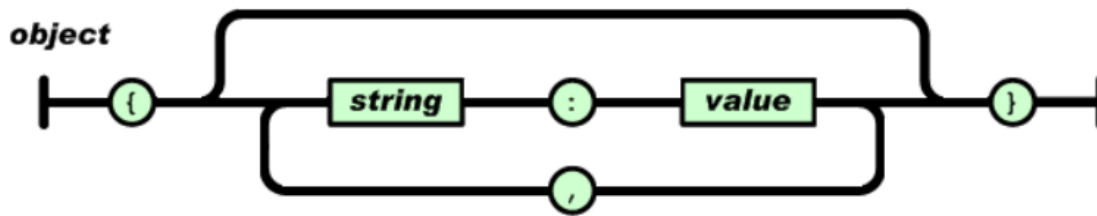
**Server-Side JavaScript:** While JavaScript is predominantly used on the client side, with the advent of technologies like Node.js, it can also be employed for server-side development.

### 1.3.3 Json

JSON (JavaScript Object Notation) is a lightweight format that is used for data interchanging. It is based on a subset of JavaScript language. It has been the preferred format because it is much more lightweight

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an object, record, structure, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.



### 1.3.7 Weather API

Weather APIs are Application Programming Interfaces that allow you to connect to large databases of weather forecast and historical information. The weather API provides enough weather data for basic weather information (eg current weather, forecast, UV index data, and historical weather information). You can use geolocation and names to get a city location.



## Chapter 2

# Problem Identification

---

### 2.1 Problem Statement

The aim of this project is to develop a comprehensive weather application website that provides users with accurate and up-to-date weather information. The website should offer a user-friendly interface, displaying real-time weather conditions, forecasts, and additional relevant data. The primary goal is to create a platform that caters to users' diverse needs related to weather information, ensuring a seamless and engaging experience.

### 2.2 Project Scope

The scope of the Weather Application Website project encompasses the development of a robust and user-friendly platform that provides comprehensive weather information. The project will include the following key components and functionalities:

|   |
|---|
| <b>User Authentication and Profiles:</b>  |
| <ul style="list-style-type: none"><li>• Implement a secure user authentication system.</li><li>• Allow users to create accounts, log in, and manage their profiles.</li><li>• Users can customize preferences, save favorite locations, and receive personalized weather updates.</li></ul> |
| <b>Real-time Weather Data:</b>  |
| <ul style="list-style-type: none"><li>• Integrate with reputable weather APIs to fetch and display accurate real-time weather information.</li><li>• Display current conditions, including temperature, humidity, wind speed, and atmospheric pressure.</li></ul>                           |
| <b>Location Services:</b>   |
| <ul style="list-style-type: none"><li>• Enabling automatic location detection for users' current weather information.</li><li>• Allowing manual input and storage of multiple locations for personalized weather tracking.</li></ul>  |








### 2.3 Design and Implementation Constraints

Design and implementation constraints refer to the limitations and restrictions that may impact the development process of the Weather Application Website. Identifying and understanding these constraints is crucial for effective project management. Here are some common design and implementation constraints for the Weather Application Website:

### **Data Accuracy and Availability:**

- Constraint: The accuracy and availability of weather data depend on third-party APIs. Inaccuracies or downtimes in these services may affect the reliability of the information provided to users.

### **API Limitations:**

- Constraint: Third-party APIs may impose limitations on the frequency and volume of data requests. The application must be designed to stay within these limits to prevent disruptions in service.

### **Technology Compatibility:**

- Constraint: Compatibility issues may arise when integrating with various devices and browsers. The design must account for differences in technology stacks and ensure a seamless experience across platforms.

## Chapter 3

# Design

---

Design" is a broad and multifaceted term that encompasses the process of creating, planning, and conceptualizing the structure and appearance of various elements to achieve a specific goal or solve a problem. The term is used across a wide range of disciplines, from visual arts and engineering to software development and business strategy. .

### **Visual Arts and Graphic Design:**

In the context of visual arts, design refers to the creation and arrangement of visual elements such as shapes, colors, images, and text to communicate a specific message or evoke a certain emotion. Graphic designers, for example, use design principles to create visually appealing and effective communication materials.

### **Product Design:**

In product design, the focus is on creating tangible items that serve a particular purpose or function. This involves considering not only the aesthetics but also the usability, functionality, and ergonomics of the product. Product designers work on everything from everyday objects to complex machinery.

### **User Interface (UI) and User Experience (UX) Design:**

In the realm of digital technology, UI design involves creating the visual elements and layout of a user interface, while UX design focuses on the overall experience a user has with a product or system. Designers aim to make interfaces intuitive, efficient, and enjoyable for users.

### **Architectural Design:**

In architecture, design refers to the process of planning and creating the layout and structure of buildings or landscapes. This includes considerations of aesthetics, functionality, safety, and sustainability.

### **Web Design:**

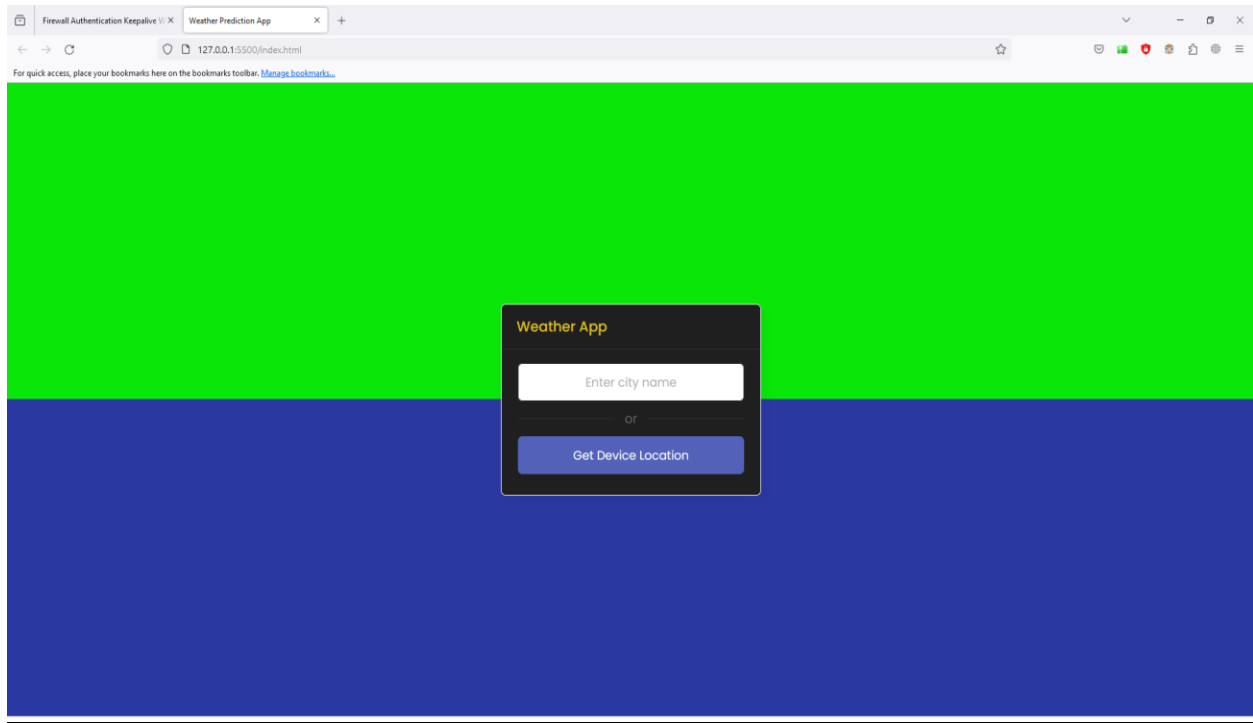
In the context of websites and online platforms, web design involves creating the visual elements, layout, and overall look and feel of a website. Web designers consider

### **3.2 Use-Case Diagram**

An interaction between a user and a system is described by use case diagram. Use case diagrams describe what a system does from the standpoint of an external observer. The emphasis is on what a system does rather than how. Use case diagrams are closely connected to scenarios. A scenario is an example of what happens when someone interacts with the system. A use case diagram is a collection of actors, use cases, and their communications.

For initial development we can use this use case. In this use case diagram we can see following use cases and actor. Use cases are self explanatory and they represent the main functions of Weather Prediction System.

## 3.3 Ui Design



## Chapter 4

# Testing

---

Testing is the process of evaluating a system or its component with the intent to find whether it satisfies the specified requirement or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. Systems should not be tested as a single, monolithic unit. The testing process should therefore proceed in the stages where testing is carried out incrementally in conjunction with system implementation. Errors in program components may come to light at a later stage of the testing process. The process is therefore an iterative one with information being fed back from later stage to earlier parts of the process. Following testing were done during the course of our project.

### 4.1 Unit Testing

Unit testing focuses verification efforts on the smaller unit of software design. Using the detailed design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of the test and the error detected as a result is limited by the constraint scope established for unit testing. The unit test is always white box oriented, and the step can be conducted in parallel for multiple modules

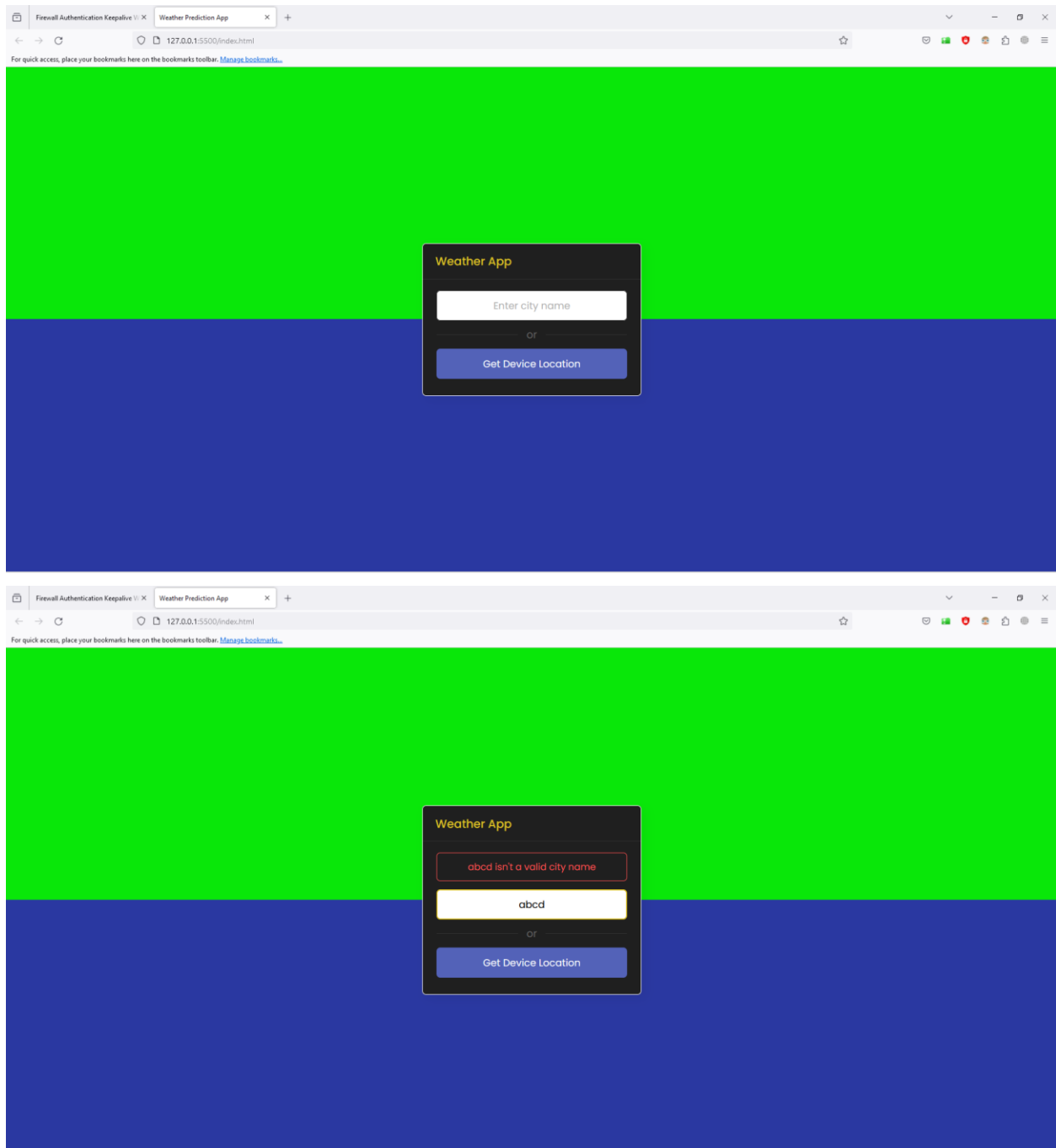
- Individual Component rendering

### 4.2 Integration Testing

With unit testing the modules may function properly, but at times they may have inadvertent affect on another, sub function when combined, may not produce the desired functions; individually acceptable impression may be signed to unacceptable levels then global data structure may present problems. Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by the design.

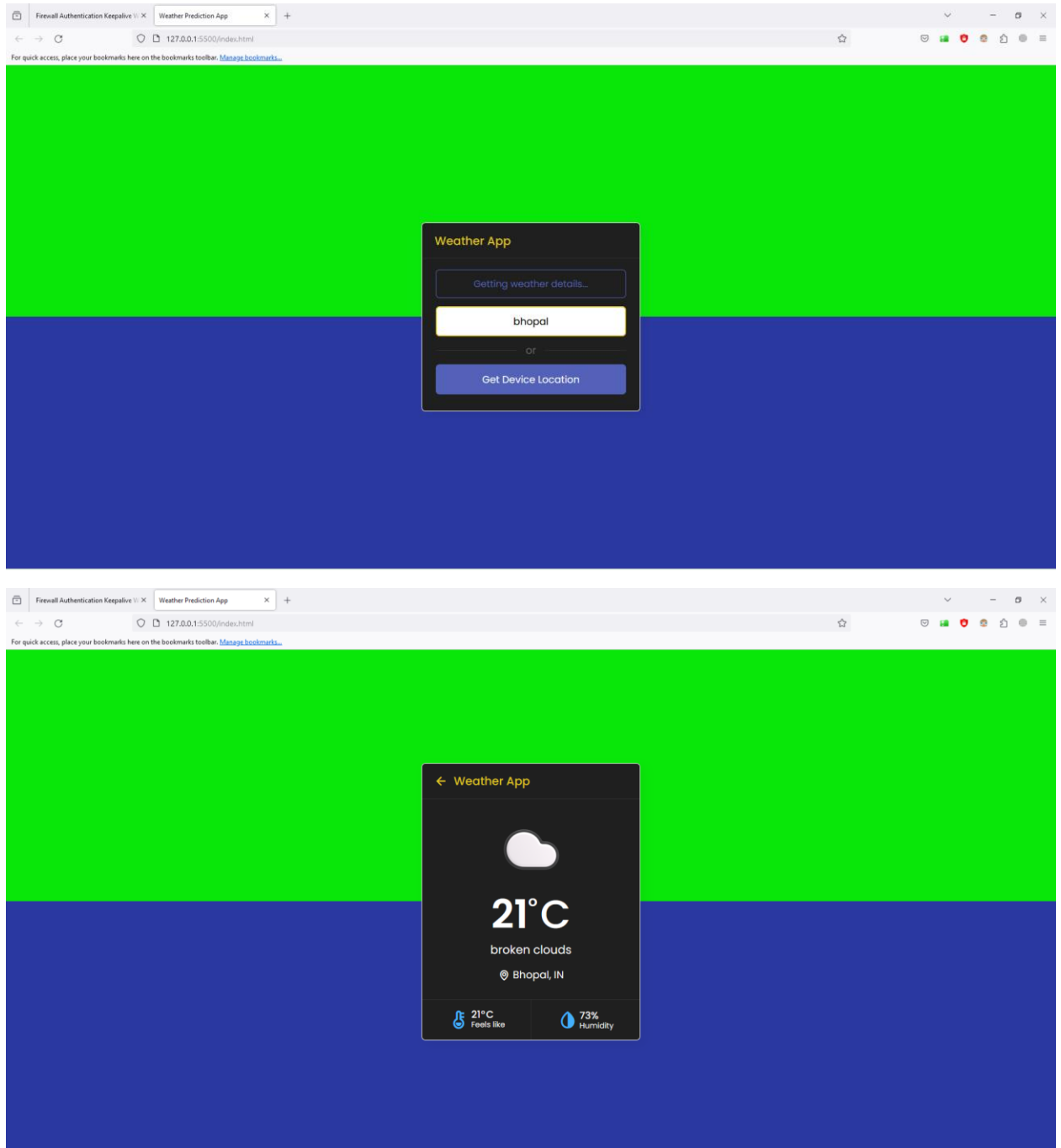
## Chapter 5 Results

### Screenshot



# Weather Prediction APP

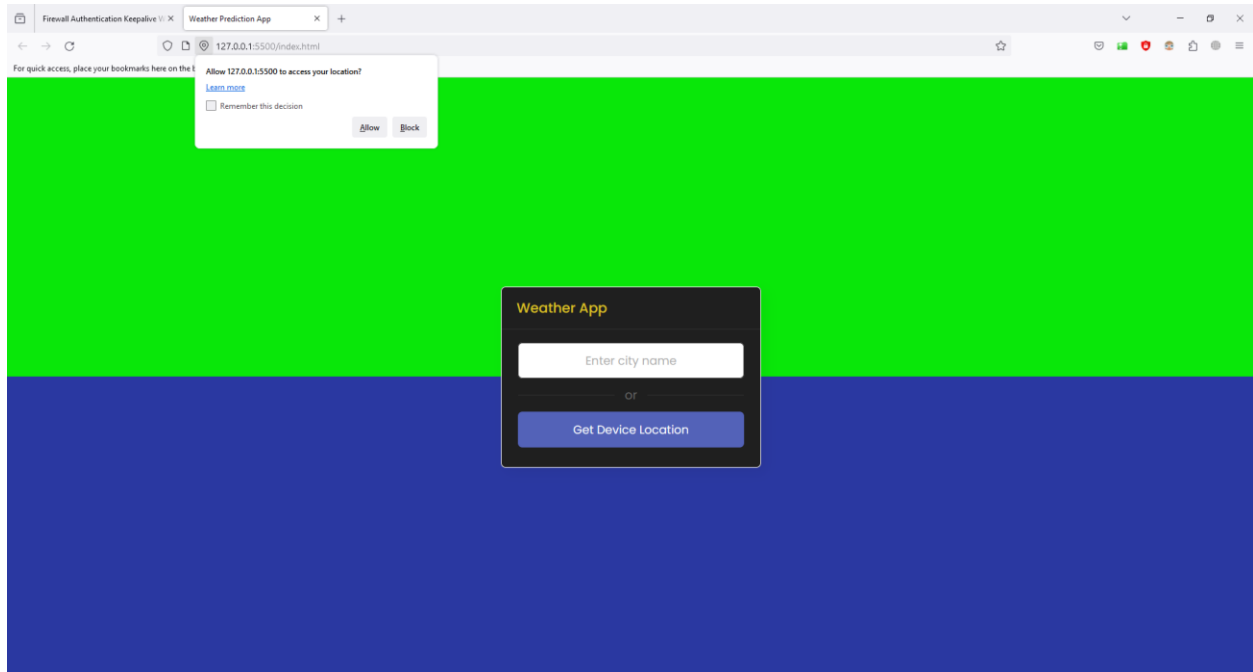
---





# Weather Prediction APP

---



## Chapter 6

# Deployment

---

### 6.1 Purpose of Deployment Phase

The deployment phase is the final phase of the software development life cycle (SDLC) and puts the product into production. After the project team tests the product and the product passes each testing phase, the product is ready to go live. This means that the product is ready to be used in a real environment by all end users of the product.

There are various phases of the deployment process the project team must follow to ensure the code and technology deploy appropriately. The phases include deployment preparation and procedures, product deployment, transferring ownership of the product, and closing the deployment phase.

### 6.2 Preparation and Procedures

In the preparation and procedures phase, the project team installs the software and conducts another test to ensure successful installation. Once the installation is complete, the project team creates operating procedures, which include instructions for how the software should work in the information technology environment. If there are issues with system functionality, the operating instructions also provide a mitigation plan to help the end user repair the issue

### 6.3 Environment Variables:

- **Visual Studio Code**-Visual Studio Code is an IDE developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open source and released under the permissive MIT License. The compiled binaries are freeware and free for private or commercial use.

Working with Git and other SCM providers has never been easier. Review diffs, stage files, and make commits right from the editor. Push and pull from any hosted SCM service. If want to add new languages, themes, debuggers, and to connect to additional services just install the extensions. Extensions run in separate processes, ensuring they won't slow down the editor.

## CONCLUSION

The design and implementation of this application have been driven by a commitment to providing users with a seamless and informative experience. Here are the key takeaways from the development and design process:

1. **User-Centric Approach:** The Weather Application Website has been developed with a user-centric approach, prioritizing the needs and preferences of users. The intuitive user interface, coupled with customizable features, ensures that users can effortlessly access the weather information most relevant to them.
2. **Real-Time and Forecast Data:** The integration of reliable weather APIs enables the website to deliver real-time weather updates and precise forecasts. Users can trust the accuracy of the information provided, empowering them to make informed decisions based on current and future weather conditions.
3. **Multifaceted Functionality:** The website offers a range of features, including interactive maps, historical weather data access, and personalized user profiles. These functionalities cater to a diverse user base, from casual users seeking quick weather updates to enthusiasts interested in in-depth weather analyses.

## **FUTURE SCOPE**

### **Scope of Weather Prediction**

- Prediction will be done based on historical weather activities like based on past temperature, wind, etc. pattern what will be the future weather.

### **Future Enhancement**

- Mobile and IOS application Integration.
- Addition of new cities weather dataset to predict there future weather also.
- Addition of new Indices.
- Animation like snow and functions like notifications can also be added.

## REFERENCES and GITHUB LINK

### References -

1. <https://www.youtube.com/watch?v=MIYQR-Ybrn4>
2. <https://www.youtube.com/watch?v=pFvWwFua6mw>
3. <https://openweathermap.org/>

### API Testing –

/https://api.openweathermap.org/data/2.5/weather?q=c(ityname)&appid=6a3cb15519a96f3d99f7aa4a2dc9ef03&units=metric

### Github Links -

1. <https://github.com/Hariomsingh01/Weather---App>

## **GUIDE LINES**

|                                  |   |
|----------------------------------|---|
| Chapter 1 Introduction           | it provides the details regarding the introduction of the area to which project belong say image processing, defines the methodology used step by step to achieve the end product, may be diagrammatically. Software tools and technology used like Java introduction and features (at the most 5-6 pages).   |
| Chapter 2 Problem Identification | problem statement, scope of the project, Objective in term of functional and non functional requirements provided by the project, top down or bottom up approach used for achieving the objective and its explanations, a formal SRS document etc. maximum number of pages 20.  |
| Chapter 3 Detail design          | it includes various UML diagrams like class diagram, use case diagrams etc., data base design, description about tables, ER diagrams, blue prints (for web based developments), flow charts, pseudo code, work breakdown structure etc.   |
| Chapter 4 Testing                | testing should be defined in tables including test case name, test case reference, description, precondition, post condition, result and remarks regarding black box testing and white box testing of unit testing, integration testing and system testing.   |
| Chapter 5 Results                | results may be in the form of tables, graphs etc along with proper screen shots and description about the screenshots like its purpose, how this screen will appear etc.  |
| Chapter 6 Deployment             | clearly defines the steps required to make settings in the pc to run your project along with the snapshot. Here by snapshot, we does not mean the project snapshot. Its about any setting made in operating system, path setting, directory where project will be stored. In short, anybody can install and run your project without your support by just referencing this chapter. |
| Conclusion                       | one page conclusion about your project that includes the summary of your work.  |
| Future Scope                     | define future improvement, suggestion that helps your juniors to extend your idea and develop the next version.   |

References details about links, books, research papers, journals that you refer during the project development.

### Note:

- All the pages except front page must be properly numbered.
- All the pages before the chapter must be numbered like i, ii, iii, iv .....
- From chapters 1 onwards, pages must be numbered as 1, 2, 3, 4 .....
- Report must be of at least 50-60 pages excluding Preface, front page, table of content, conclusion, future scope, appendix and references.
- Chapter heading Font size 20, Bold, Right aligned, Times new roman, no spacing and a complete line below.
- Paragraph heading font size 14, Bold, Left aligned, Times new Roman, spacing 6 pts above and below, Line spacing 1.5, not underlined.
- Paragraph content font size 12, justified, Times new roman, 6 pts above and below, 1.5 line spacing, not underlined.
- Number of copies  $N+3$  spiral bound, where  $N$  is number of students in group, 2 for department and 1 for project supervisor (in needed by him/her)
- Bring one copy during internal viva and must be checked and approved by supervisor.
- If approved then bring rest of the copies during External viva.
- Margin
  - Left : 1.25 inch
  - Right : 1 inch
  - Top : 1 inch
  - Bottom : 1 inch
- Figures and followed by their name must be center aligned.
- Figures must me numbered as per the chapter number and must be used in the chapter. For example, if there is a third figure of Computer Block diagram in chapter 2 then it must be named as

#### **Figure 2.3: Computer Block Diagram**

- Same is the case with any table used.
- Conclusion, future scope, references, appendix A and appendix B will not be mentioned as a chapter, Font Size 20, bold, center aligned, Times New Roman, spacing 12pts above and below, not underlined.
- Use all figures, tables, screen shots properly in explanation properly like as shown in figure 2.3.