**PROJECT REPORT ON**

**WEATHER FORECASTING WEBSITE**

**Submitted under partial fulfillment of the requirements to**

**IT 353 – PROJECT 1**

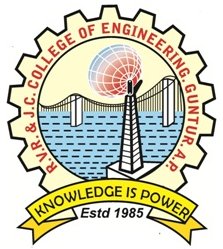
By

N. SRI HARSHA (Y21IT087)

P. MANI KANTA SWAMY (Y21IT102)

SK. ALTHAF ALI MUBARAK (Y21IT109)

T. SAI KUMAR REDDY (Y21IT122)



**OCTOBER 2023**

**Department of Information Technology**

**R.V.R. & J.C. College of Engineering**

**(Affiliated to Achary­­a Nagarjuna University, Guntur)**

i

# **R.V.R. & J.C. COLLEGE OF ENGINEERING**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**BONAFIDE CERTIFICATE**

This is to certify that this project work titled WEATHER FORECASTING WEBSITE is the bonafide work of (N. Sri Harsha (Y21IT087), P. Mani Kanta Reddy (Y21IT102), Sk. Althaf Ali Mubarak (Y21IT109), T. Sai Kuma Reddy (Y21IT122)) who have carried out the work under my supervision, and submitted in partial fulfillment of the requirements to **IT-353, PROJECT 1** during the year 2023-2024.

**Dr. A. Yaswanth Kumar Dr. A. Srikrishna**

Assistant Professor Prof. & HOD, Dept. of IT

ii

## ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without proper suggestions, guidance, and environment. The combination of these three factors acts as the backbone of our Project **“WEATHER FORECASTING WEBSITE”.**

We would like to express my gratitude to the Management of **R.V.R. & J.C. COLLEGE OF ENGINEERING** for providing me with a pleasant environment and excellent lab facility.

We are very much thankful to our Principal, **Dr. Kolla Srinivas,** for providing support and a stimulating environment.

We express our sincere thanks to **Dr. A. Srikrishna,** Professor, and Head of the Department of Information Technology, for her valuable suggestions during the mini project.

We are very glad to express our special thanks to our guide **Dr. A. Yaswanth Kumar,** Assistant Professor in the Department of Information Technology for his timely, guidance and for providing us with the most essential materials required for the completion of this project.

Finally, we express our sincere thanks to all the **Teaching** and **Non-Teaching** staff of the **IT Department** who have contributed to the successful completion of this report.

N. SRI HARSHA (Y21IT087) P. MANI KANTA SWAMY (Y21IT102) SK. ALTHAF ALI MUBARAK (Y21IT109) T. SAI KUMAR REDDY (Y21IT122)

iii

**ABSTRACT**

Weather forecasting is always needed by a person daily, for example, you would already be aware of a rainstorm hours before it would arrive. So, to display weather forecasts on my website, I would connect server data i.e., an **API** (An application program interface (**API**) is a set of routines, protocols, and tools for building software applications. An **API** specifies how software components should interact. Additionally, **APIs** are used when programming graphical user interface (GUI) components) through which I would access its data and connect it with my website through my **API** key. I would use open weather **API** for this project through which I will access all its real-time weather data and display it on my web page using Javascript, CSS, and HTML.

iv **TABLE OF CONTENTS**

**Chapter No. & Name Page No.**

**1. Introduction**

1.1 Outline of the Project 1.2 Literature Review 1.3 Problem Statement 1.4 Objectives

**2. Aim & Scope**

2.1 Requirements

2.1.1 Hardware Requirements

2.1.2 Software Requirements

2.2 Role of Editor

2.3 Languages Used

2.3.1 HTML

2.3.2 CSS

2.3.3 JavaScript

2.4 Open Weather API

2.4.1 Current Weather Data

2.4.2 Forecast

2.4.3 Searching

**3. UML Diagrams**

3.1 Use Case Diagrams

3.1.1 Identification of Actors, Use Cases, Relations

3.1.2 Construction of Use Case Diagram, Flow of Events

3.2 Sequence Diagram & its Construction 3.3 Class Diagram & its Construction

**4. Methods & Material Used**

3.1 Designing of Text Editor

3.2 Connection of API

v

**5. Result & Discussion**

**6. Conclusion and Future Work** 6.1 Summary 6.2 Future Work

**Appendix A.** Source Code **B.** Screenshots

**References**

vi

**LIST OF FIGURES**

**Figure Name Page No.**

Outline of the Webpage Styling of the Webpage API connection through Java script Open Weather API API key Weather application Chennai Delhi London

vii

**LIST OF ABBREVIATIONS**

**ABBREVIATION EXPANSION**

HTML Hyper Text Mark Up Language

CSS Cascading Style Sheet

JS JavaScript

API Application Program Interface

viii

**CHAPTER 1**

**INTRODUCTION**

**Hypertext Markup Language (HTML)** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delimited by tags, and written using angle brackets. Tags such as <img /> and <input /> directly introduce content to the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**API**

In computer programming, an **application programming interface (API)** is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components. A good API makes it easier to develop a computer program by providing all the building blocks, which are then put together by the programmer.

An API may be for a web-based system, operating system, database system, computer hardware, or software library.

An API specification can take many forms but often includes specifications for routines, data structures, object classes, variables, or remote calls. POSIX, Windows API, and ASPI are examples of different forms of APIs. Documentation for the API usually is provided to facilitate usage and implementation.

1

**1.1. OUTLINE OF THE PROJECT**

Throughout human history, people have been keen to know about the weather, its parameters, and its impacts on their daily lives. By virtue the technological advancement, in this era, the information about the weather lies in our hands (through mobile phones or websites). We can now make ourselves aware of not only our location's temperature but also any part of the world. In this project, we will learn how to make a weather application using JavaScript.

As a web developer, grabbing data from APIs is something you are going to do often. Fetching weather data is a perfect way to get your feet wet. In this project, we are going to use the browser's built-in fetch with JavaScript to grab data from Open Weather Map's API.

**1.2. LITERATURE REVIEW**

**JavaScript**, often abbreviated as **JS**, is a high-level, interpreted scripting language that conforms to the ECMA Script specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it, and major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

2

**1.3. PROBLEM STATEMENT**

Develop a user-friendly website that provides accurate temperature forecasts for cities around the world using the OpenWeather API. The website should allow users to search for a city and view its current temperature, along with a 5-day forecast. The interface should be intuitive, responsive, and visually appealing across various devices. Ensure the website fetches real-time data from the OpenWeather API and presents it in a clear and understandable format for users.

**1.4. OBJECTIVES**

First, we need to outline the project using HTML and CSS to create the front end of the web page.

3

**CHAPTER 2**

**AIM & SCOPE OF WEB DEVELOPMENT**

**2.1. REQUIREMENTS:**

**2.1.1. HARDWARE REQUIREMENTS:**

1. Any processor with minimum GPU memory

**2.1.2. SOFTWARE REQUIREMENTS:**

**1.** Visual Studio Code Editor

**2.** Open Weather API

**2.2. ROLE OF VISUAL STUDIO CODE:**

**Visual Studio Code**, also commonly referred to as **VS Code**, is a source-code editor developed by Microsoft for Windows, Linux, and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, and preferences, and install extensions that add functionality.

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 [Build](https://en.wikipedia.org/wiki/Build_(developer_conference)) conference. A [preview](https://en.wikipedia.org/wiki/Technical_preview) build was released shortly thereafter.

On November 18, 2015, the [source code](https://en.wikipedia.org/wiki/Source_code) of Visual Studio Code was released under the [MIT License](https://en.wikipedia.org/wiki/MIT_License) and made available on [GitHub](https://en.wikipedia.org/wiki/GitHub). Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the [public preview](https://en.wikipedia.org/wiki/Beta_software) stage and was [released to the web](https://en.wikipedia.org/wiki/Software_release_life_cycle#Web_release). Microsoft has released most of Visual Studio Code's source code on [GitHub](https://en.wikipedia.org/wiki/GitHub) under the permissive [MIT License](https://en.wikipedia.org/wiki/MIT_License), while the binary releases by Microsoft are [freeware](https://en.wikipedia.org/wiki/Freeware), and include [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) code.

Visual Studio Code is a source code editor that can be used with a variety of programming languages, including C, CSS, C++, HTML, Java, JavaScript, Node.js, and Python. It is based on the Electron framework, which is used to develop Node.js web applications that run on the Blink layout engine.

4

**LANGUAGES USED**

* **HTML**
* **CSS**
* **JAVASCRIPT**

**HTML**

Hypertext Markup Language (HTML) is a standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from the web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of the web page semantically and originally included cues for the appearance of the document.

**CSS**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, and enable multiple web pages to share formatting by specifying the relevant CSS separately. CSS file, and reduce complexity and repetition in the structural content.

**JAVA SCRIPT**

**JavaScript** often abbreviated as **JS**, is a high-level, interpreted scripting language that conforms to the ECMA Script specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object orientation, and first-class functions. Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications.

5

**2.3 OPEN WEATHER API**

Open Weather Map is an online service that provides weather data, including current weather data, forecasts, and historical data to the developers of web services and mobile applications. For data sources, it utilizes meteorological broadcast services, raw data from airport weather stations, raw data from radar stations, and raw data from other official weather stations. All data is processed by Open Weather Map in a way that it attempts to provide accurate online weather forecast data and weather maps, such as those for clouds or precipitation. Beyond that, the service is focused on the social aspect by involving weather station owners in connecting to the service and thereby increasing weather data accuracy. The idealogy is inspired by Open Street Map and Wikipedia which makes information free and available for everybody. It uses the Open Street Map for the display of weather maps.

Open Weather Map provides an API with JSON, XML, and HTML endpoints and a limited free usage tier. Making more than 60 calls per minute requires a paid subscription starting at USD 40 per month.

Access to historical data requires a subscription starting at US$150 per month.

Users can request current weather information, extended forecasts, and graphical maps (showing cloud cover, wind speed, pressure, and precipitation).

**CURRENT WEATHER DATA**

Current data is refreshed every ten minutes; it can be searched by city or by geographic coordinates on Earth.

**FORECASTS**

Weather forecasts can be searched by city or by coordinates. Three-hourly forecasts are available for up to 5 days, while daily forecasts are available for up to 16 days.

**PROBLEM STATEMENT**

Develop a user-friendly website that provides accurate and up-to-date temperature forecasts for cities around the world using the OpenWeather API. The website should allow users to search for a city and view its current temperature, along with a 5-day forecast. It should also display additional weather details such as weather conditions. The interface should be intuitive, responsive, and visually appealing across various devices. Ensure the website fetches real-time data from the OpenWeather API and presents it in a clear and understandable format for users. 6

**SEARCHING**

The Open Weather Map geocoding system allows users to select cities by name, country, zip - code, or geographic coordinates. It is possible to search by part of the city name. To make searching result more accurate city name and country should be divided by a comma.

**MAPS**

Open Weather Map service provides lots of weather maps including Precipitation, Clouds, Pressure, Temperature, Wind, and many others. Maps can be connected to mobile applications and websites. Weather maps can be connected as layers to a wide range of maps including Direct tiles, WMS, Open Layers, Leaflet, Google maps, and Yandex maps.

7

**CHAPTER 3**

**UML DIAGRAMS**

**3.1 Use Case Diagrams**

A **use case diagram** is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

While a [use case](https://en.wikipedia.org/wiki/Use_case) itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a higher-level view of the system. It has been said before that “Use case diagrams are the blueprints for your system”.

Due to their simplistic nature, use case diagrams can be a good communication tool for [stakeholders](https://en.wikipedia.org/wiki/Project_stakeholder). The drawings attempt to mimic the real world and provide a view for the [stakeholder](https://en.wikipedia.org/wiki/Project_stakeholder) to understand how the system is going to be designed.

**3.1.1 Identification of Actors, Use Cases, Relations**

**Identification of Actors:**

Actors represent system users. They are not part of the system. They represent anyone or anything that interacts with the system.

An actor is someone or something that:

* Interacts with or uses the system
* Provides input to and receives information from the system
* Is external to the system and has no control over the use cases

Actors are discovered by examining:

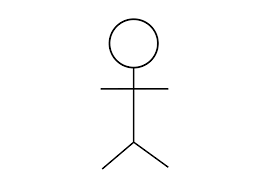
* Who directly uses the system
* Who is resposible for maintaining the system
* External hardware used by the system
* Other systems that need to interact with the system

The needs of the actor are used to develop use cases. This ensures that the system will be what the user expected.

8

**Graphical depiction:**

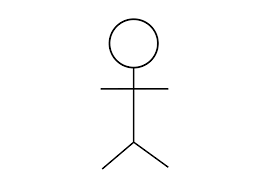
An actor is a stereotype of a class and is depicted as a “stickman” on a use-case diagram. For example,

 **Actor**

Actors identified in the information system are:

1. User: User is allowed to use the website provided

* to enter the location
* to view the weather conditions along with hourly, day forecasts

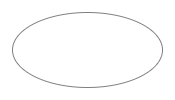
 **User**

**Identification of Use-Cases or Sub Use-Cases**

Use case can be described as a specific way of using the system from a user’s perspective. A more detailed description might characterize a use case as:

* A pattern of behaviour the system exhibits
* A sequence of related transactions performed by an actor and the system

The UML notation for use case is:



9

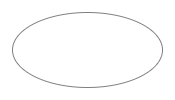
**Purpose of Use Cases:**

* Well structured use cases denote essential system or subsystem behaviours only, and are neither overly general nor too specific.
* A use case represents a functional requirement of the system as a whole
* Use cases represent an external view of the system
* A use case describes a set of sequences, in which each sequence represents the interaction of the things outside the system with the system itself.

**Use -cases identified for Weather Forecasting Website are:**

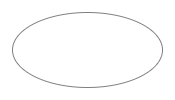
1. **Use-case name:** Search Location

This is the use case which is used by the actor to search a location and view the Weather given by Open Weather API for that particular location.

 Search Location

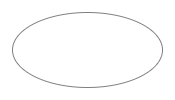
**2. Use-case name:** View Current Weather

This is the use case which shows the current weather conditions to the user.

 View Current Weather

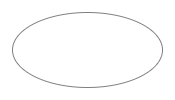
**3. Use-case name:** View Upcoming Forecasts

This is the use case that allow user to see the upcoming forecasts for next 5 different time slots. Note that the difference between two successive time slots is 3 hours.

 View Upcoming Forecasts

10

**4. Use-case name:** View Next 4 days Forecast

This is the use case that allow user to see the upcoming forecasts for next 4 days.  View Next 4 days Forecast

**Identification of Relations**

**Association Relationship:**

An association provides a pathway for communication. The communication can be between use cases, actors, classes or interfaces. If two objects are usually considered independently, the relationship is an association.

 **User Login**

**Dependency Relationship:**

A dependency is a relationship between two model elements in which a change to one model element will affect the other model element. Use a dependency relationship to connect model elements with the same level of meaning.

We can provide here:

1. Include relationship:

It is a stereotyped relationship that connects a base use case to an inclusion use case. An include relationship specifies how the behaviour in the inclusion use case is used by the base use case.

<<include>>

--------------------------->

Base use-case Inclusion use-case

11

2. Extend relationship:

It is a stereotyped relationship that specifies how the functionality of one use case can be inserted into the functionality of another use case. <<extend>> is used when you wish to show that a use case provides additional functionality that may be required in another use case.

<<extend>>

🡨------------------------------------

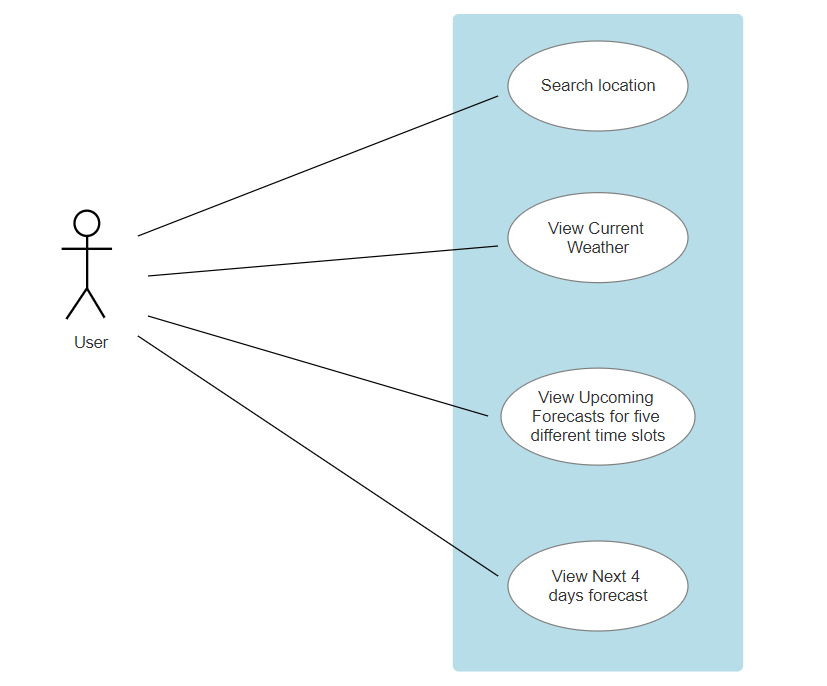
Print campaign summary check campaign budget

**3.1.2 Construction of Use Case Diagram and Flow of Events**

Use-case diagrams graphically represent system behavior. These diagrams present a high level view of how the system is used as viewed from an outsider’s perspective.

Use-case diagrams can be used during analysis to capture the system requirements and to understand how the system should work. During the design phase, you can use use-case diagrams to specify the behavior of the system as implemented.

**Use case diagram for Weather Forecasting Website:**



12

**Flow of Events**

A flow of events is a sequence of operations performed by the system. They typically contain very detailed information. Flow of events document is typically created in the elaboration phase.

Each use case is documented with flow of events

* A description of events needed to accomplish required behaviour
* Written in terms of what the system should do, not how it should do
* Written in the domain language, not in terms of the implementation

A flow of events should include

* When and how the use case starts and ends
* What interaction the use case has with the actors
* What data is needed by the use case
* The description of any alternate or exceptional flows

The flow of events for a use case is contained in a document called the use case specification. Each project should use a standard template for the creation of the use case specification. Includes the following

1. Use case name - Brief Description

2. Flow of events –

1. Basic flow

2. Alternate flow

3. Special requirements

4. Pre-conditions

5. Post-conditions

6. Extension points

**Flow of Events for finding Weather Conditions:**

**1. Use Case:** Search location

Brief Description: This use case is started by User. It provides the capability for the user to enter the location.

**2. Actor:** User

**3. Flow events:**

**3.1 Basic flow:**

* This use case begins when the website is opened and enters location in the search text box.
* If the searched location not exist the alternate flow 3.2 is executed.

13

**3.2 Alternative flow:**

* If user enters the location that does not exist in the world, it does not give the clear output that is expected by the user.

**4. Pre-condition:** User have to enter the location that exists in the world.

**5. Post-condition:** If the Website did not give the clear output, he gave the wrong location and have to change the location that exists in the world.

**CHAPTER 4**

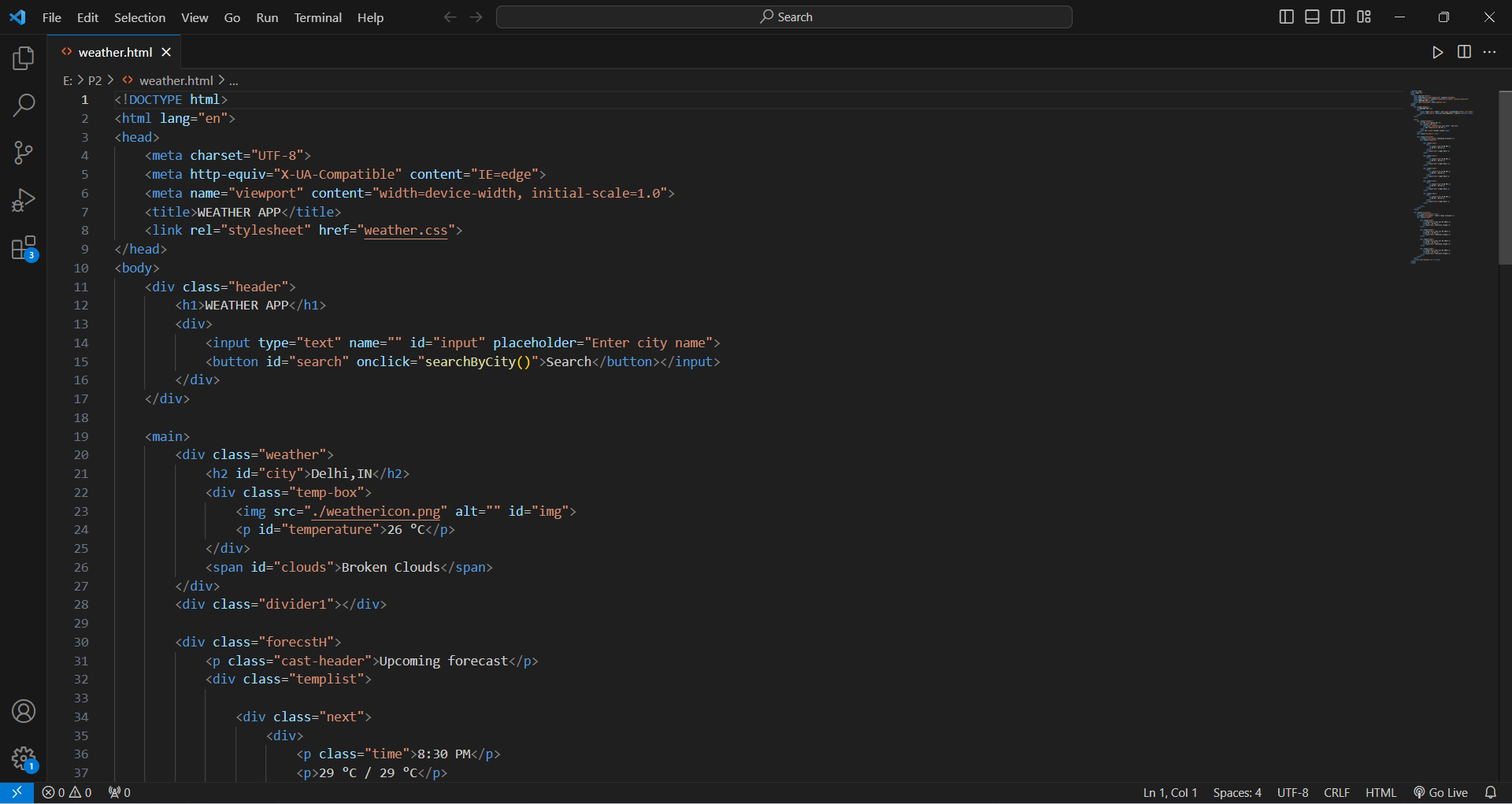
**METHODS & MATERIAL USED**

**MATERIAL USED:**

* Visual Studio Code Editor
* API Key
* Google Chrome/Microsoft Edge/Firefox

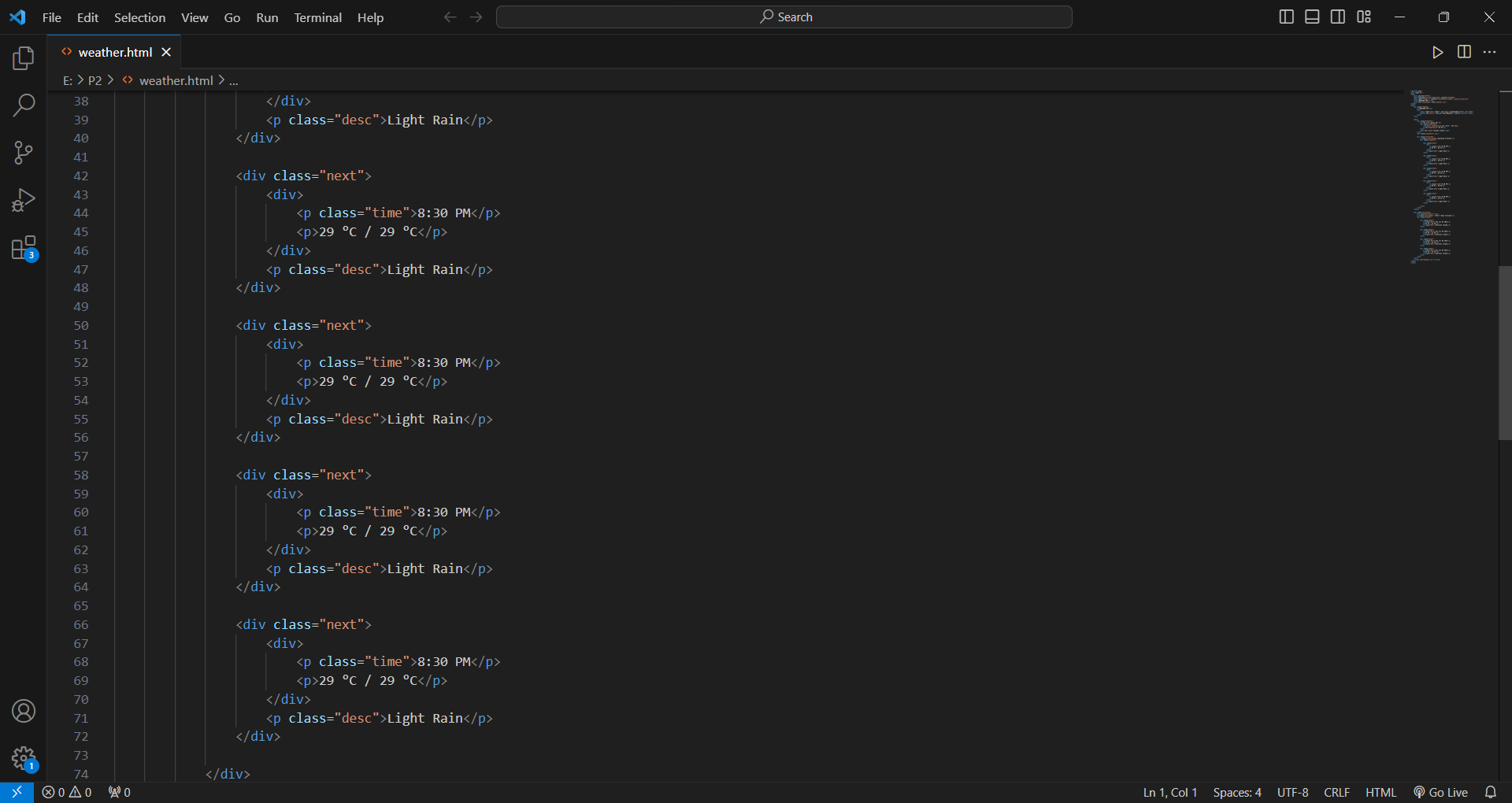
**DESIGNING IN TEXT EDITOR:**

The project will stick to the basic functionalities expected of a simple text editor which includes the ability to write something on the notepad, save it and open and modify it whenever required. For the purpose of this tutorial we will make use of VS Code editor to html and css code to create the front end part of the web page.



**Fig. 3.1 OUTLINE OF THE PROJECT**

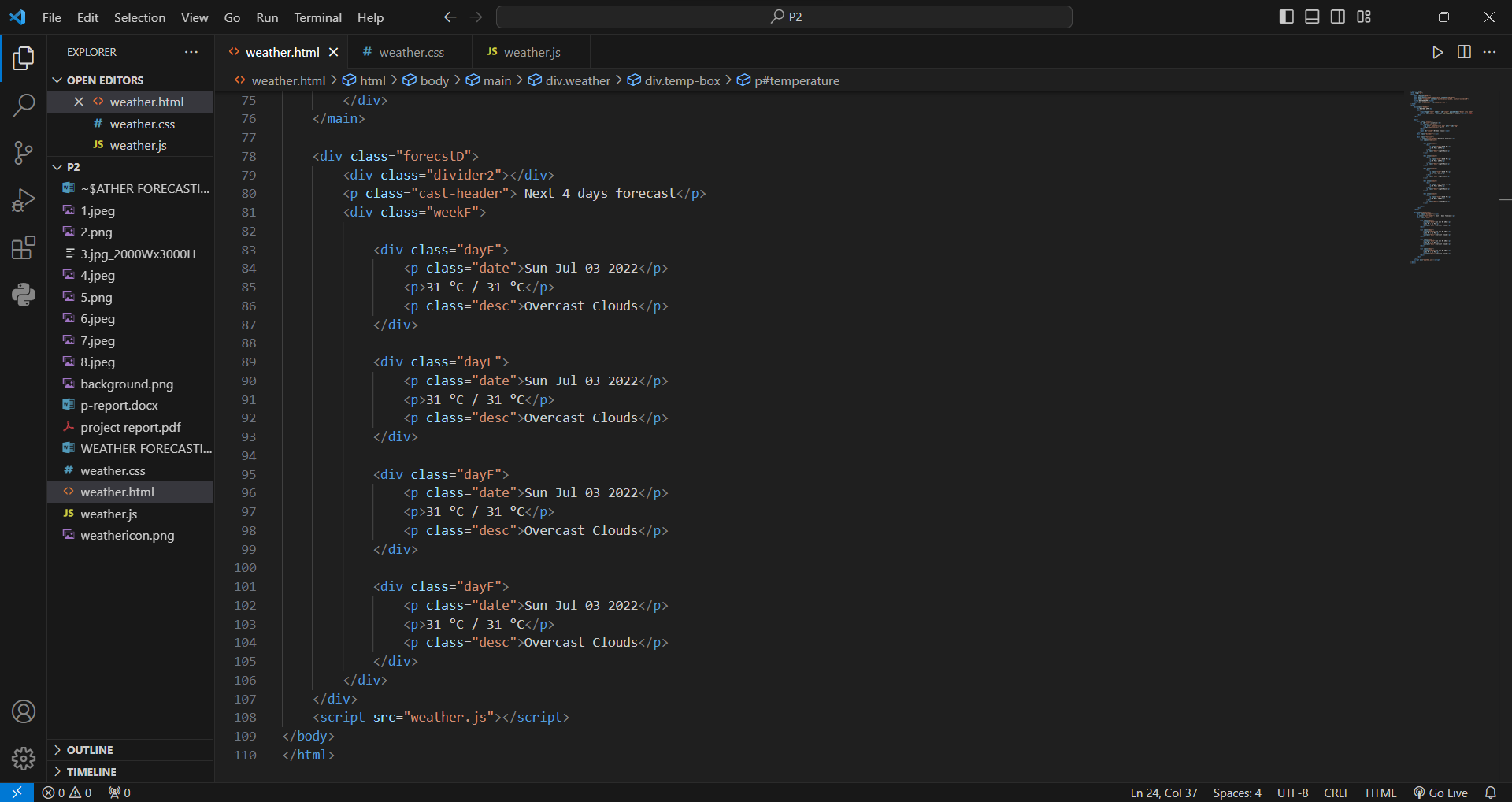
At this point we have already looked at some HTML code, how to place heading, how to add text box, how to link CSS file to the HTML page (Note that both HTML and CSS files are in the same directory), how to add button, and how different tags (like div, span, etc.) works. Now it’s time to look at the next part of the HTML page which includes adding weather forecasts to the static webpage.



**FIG. 3.2 TODAY’S FORECASTING PAGE**

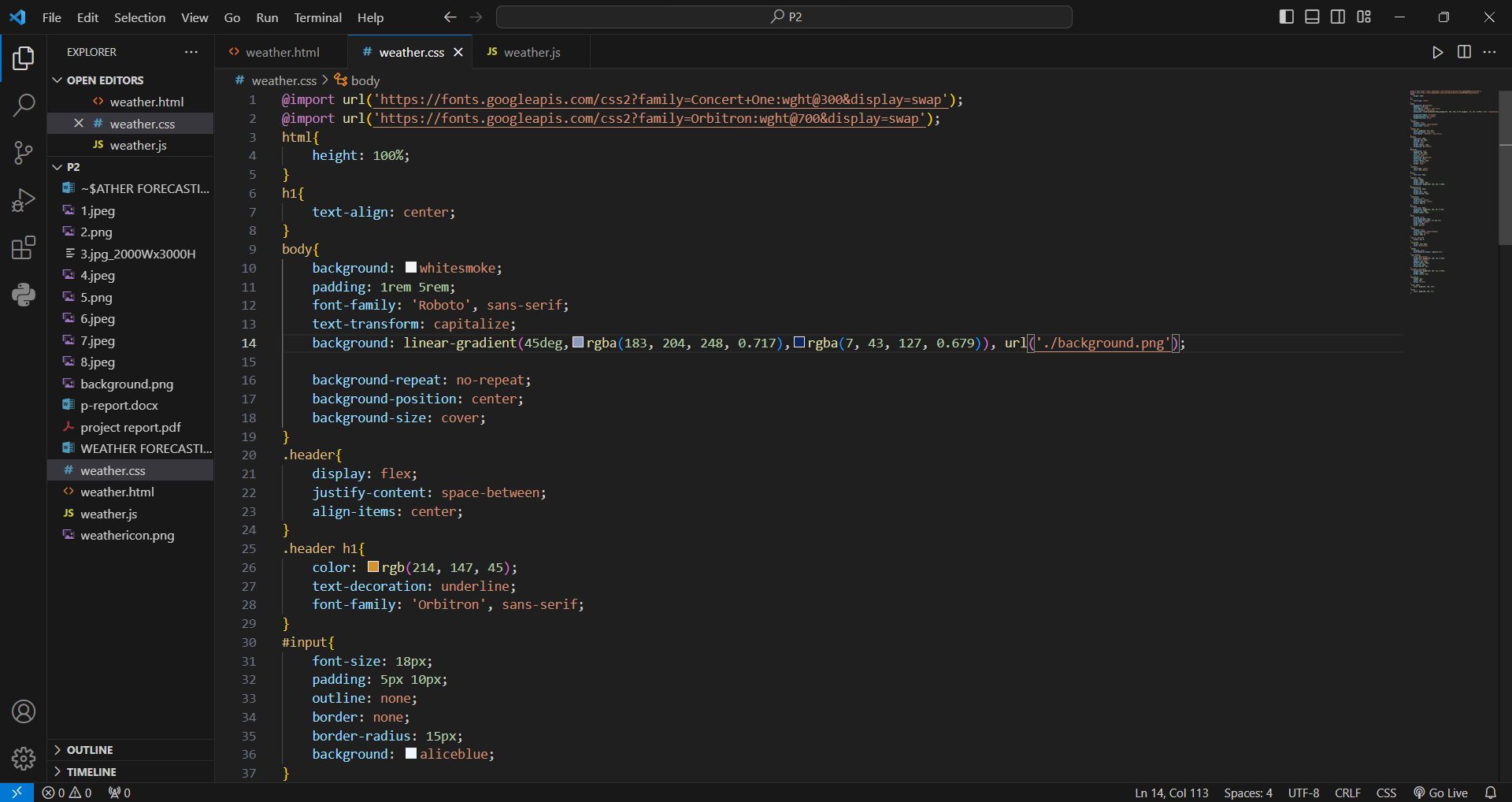
In the above figure we had observed that our **“WEATHER FORECASTING WEBSITE”** include the range of today’s weather in 5 different time slots. The time slots depends on the time of searching. For example if we searched at 7:00 PM, it shows the weather for 8:30 PM, 11:30 PM, 2:30 AM, 5:30 AM, 8:30 AM. If we searched at 7:00 AM, then it shows weather for 8:30 AM, 11:30 AM, 2:30 PM, 5:30 PM, 8:30 PM. It prints the temperature in °C along with the weather conditions (like Light Rain, Overcast Clouds, Broken Clouds, Scattered Clouds, Clear Sky, Few Clouds etc.).

In the next figure when we observe that our website include the range of next four days forecast from the day of search.

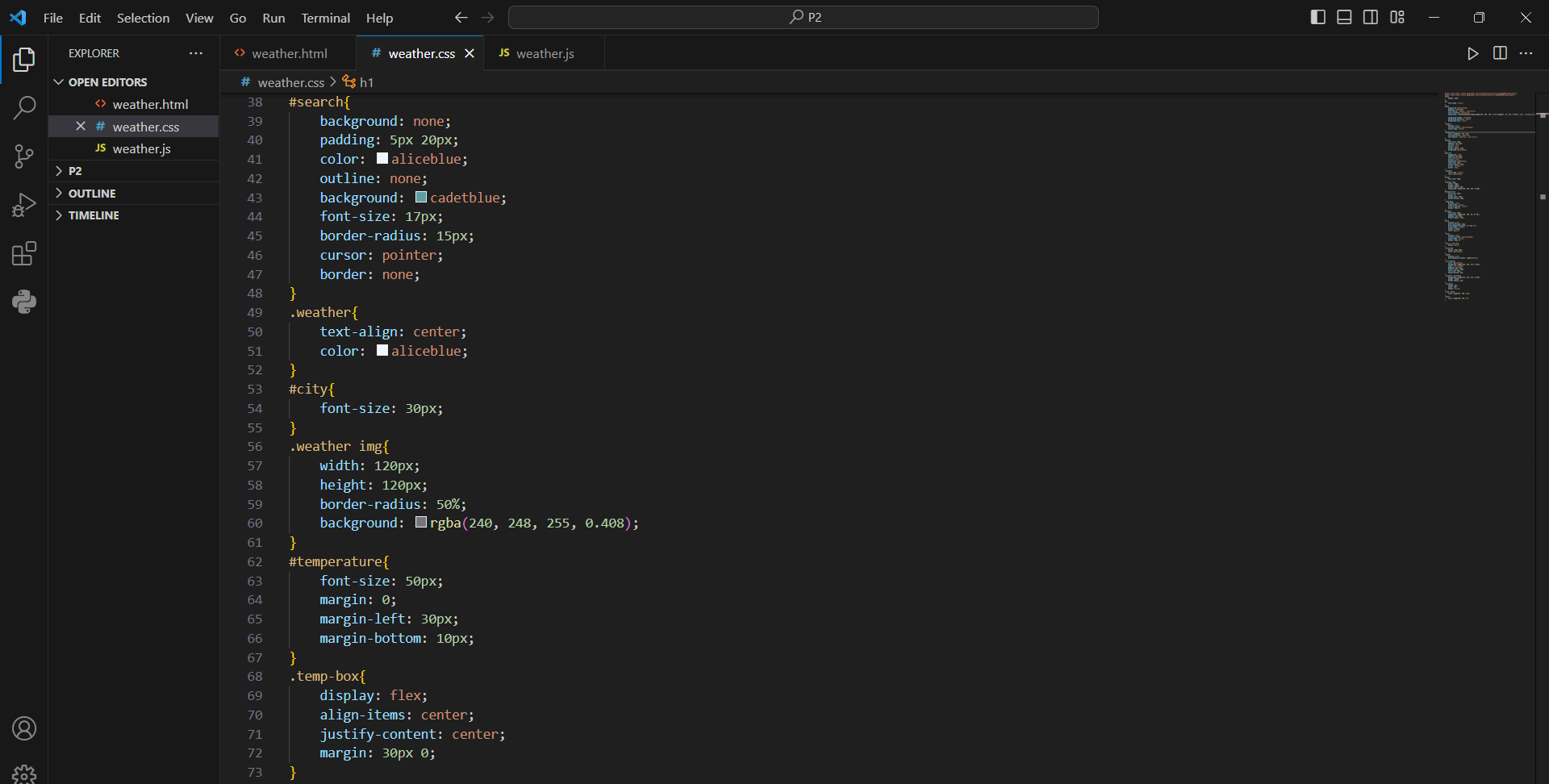


**FIG. 3.3 NEXT 4 DAYS FORECAST**

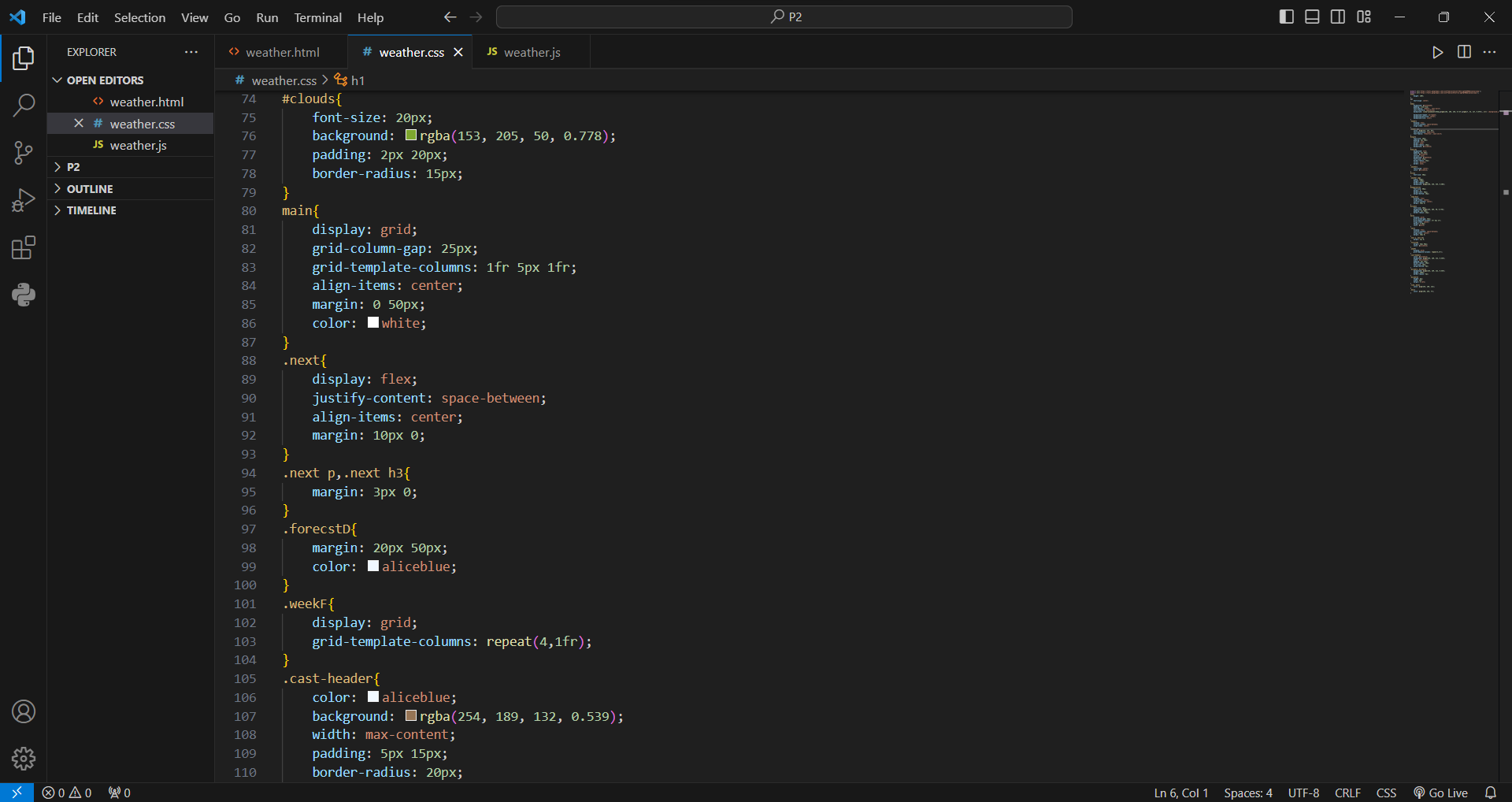
Now this is the time to look at CSS styling, the below figure describes how to style different elements like headings (h1, h2, h3, h4, h5, h6), body, etc. CSS allows us to apply styles through “id’s”, and “classes”. To apply styling through “id” we can use “#” symbol and for applying styles through “class” we can use “.” symbol.

 **FIG. 3.4 STYLING OF THE WEEB PAGE**

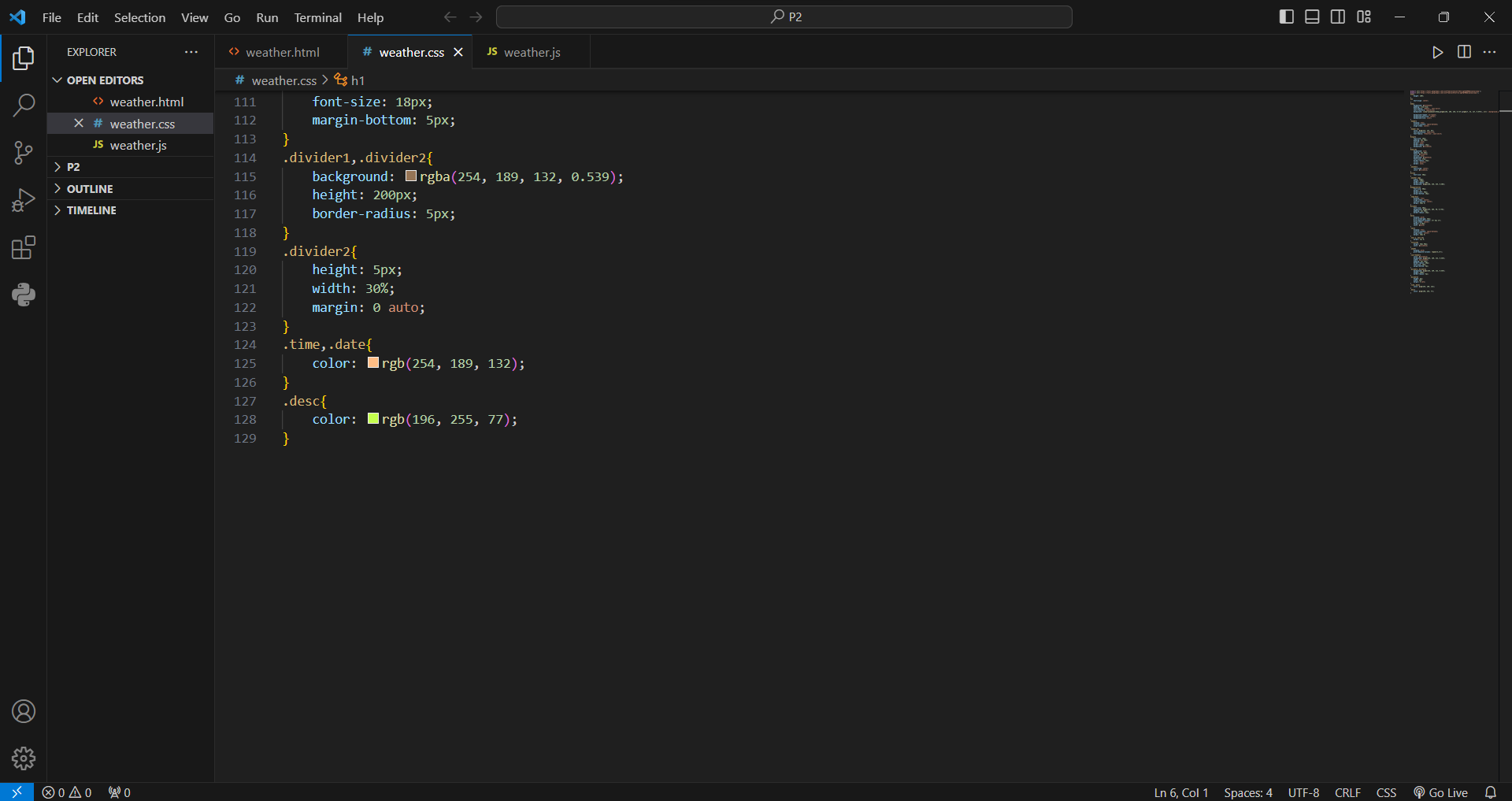
In Fig. 3.5, we applied sytles using id’s(#) and classes(.). For this we have to include id or class attribute in the element we want to style. Note that no two elements have the same id, and CSS allow any number of elements to use same class name. But while applying styles using classes, if we want to apply style to the particular element having the class name same as another element’s class name, use the class name followed by element name in a format of “.class\_name element\_name”.

 **FIG. 3.5 CSS USING ID’S AND CLASSES**

Below figures show how to apply styles to more than one element simultaneously. To do this use the elements separated by comma to apply same styles.

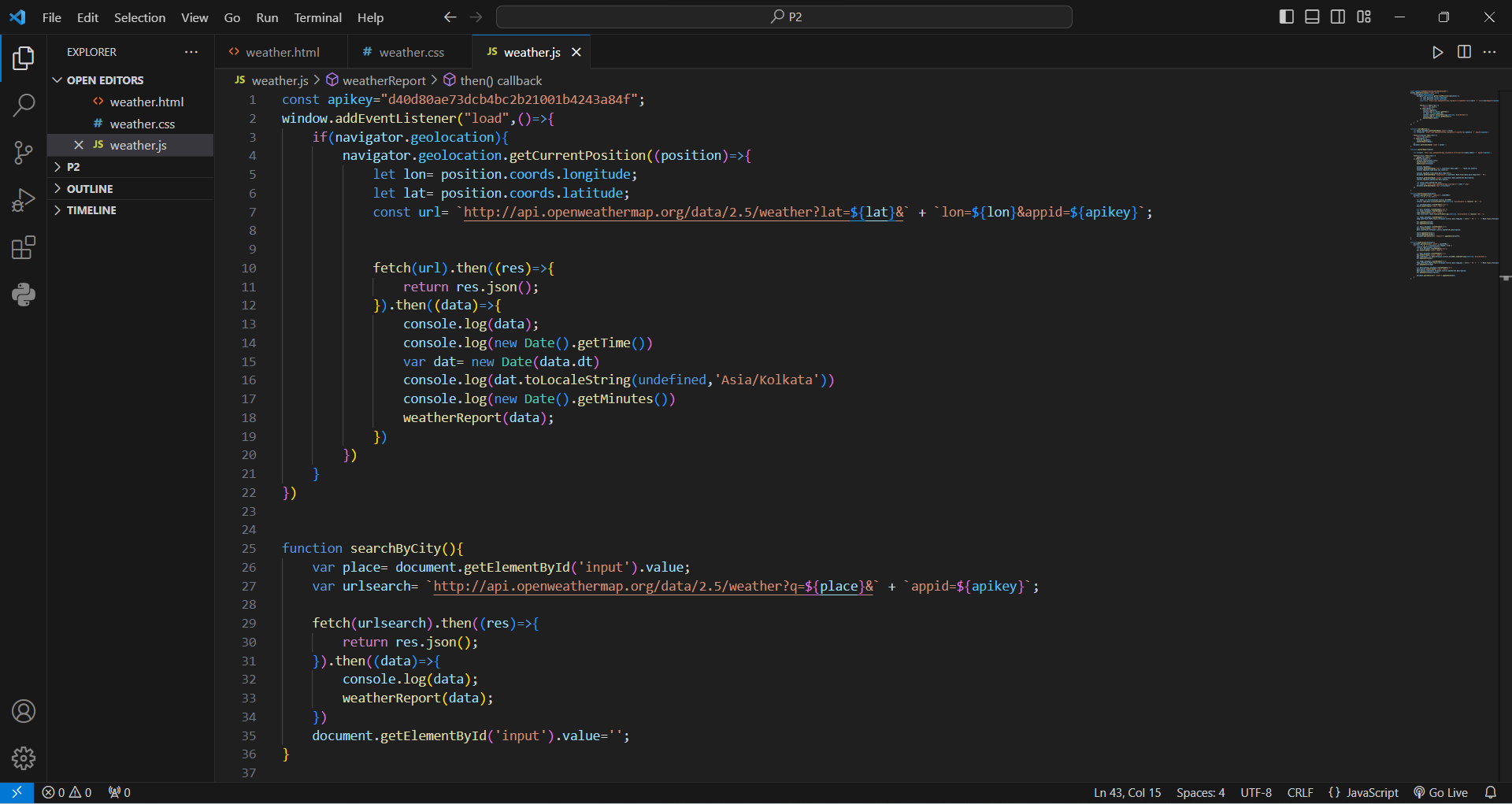
****

**FIG. 3.6 APPLY STYLES TO MORE THAN ONE ELEMENT**



**FIG. 3.7 CSS STYLING USING CLASSES**

The above figure describes applying of styles using classes. At the same time, we also applied same style simultaneously to the two elements.



**FIG. 3.8 INCLUDING API KEY IN JS FILE**

In the first line of above figure, API key is stored in “apikey” variable to access the weather. Using that key we can access temperature, humidity, weather conditions, etc.