**FETCHING WEATHER INFO**

**A Project Report Submitted in partial fulfilment of the requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

**Submitted By**

**Sishir Bohara 18221A05B9**

**Rupali Adhikari 18221A05C0**

**Devram Yadav 18221A05B8**

**Under the Supervision of**

**Dr. B S N Murthy**

**M.Tech, Ph.D.**

**Associate Professor**

**­**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE**

**(AUTONOMOUS)**

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)

(Accredited by N.B.A & NAAC with ‘A’ Grade)

**ODALAREVU – 533210**

**2018-22**

**BONAM VENKATA CHALAMAYYA ENGINEERING COLLEGE**

**(AUTONOMOUS)**

(Approved by A.I.C.T.E, New Delhi & Permanently Affiliated to J. N.T.U.K, Kakinada)

(Accredited by N.B.A & NAAC with ‘A’ Grade)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the project work entitled **“FETCHING WEATHER INFORMATION”** is being submitted for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**, at **BVC Engineering College**, **Odalarevu**, is a bonafide work done by **Sishir Bohara (18221A05B9), Rupali Adhikari (18221A05C0), Devram Yadav (18221A05B9)** under the academic year 2021-22 and it has been found suitable for acceptance according to the requirement of University. The results embodied in this thesis have not been submitted to any other University Institute for the award of any degree.

**Head of the Department**

**Dr. G Jena**

**M.Tech. Ph.D**

**Professor**

**Project Guide**

**Dr. B S N Murthy**

**M.Tech. Ph.D**

**Associate Professor**

**External Examiner**

**ACKNOWLEDGEMENT**

We express our deep sense of gratitude and respect towards our guide   
**Dr. B S N Murthy** **Associate Professor** of CSE Department, BVC Engineering College, Odalarevu for his excellent guidance right from selection of the Project work. Her constant encouragement and support have been the cause of our success in completing this thesis in the college.

We are thankful To **Dr. GUNAMANI JENA**, **Professor and Head of the Department of CSE** for facilitating, Training, providing us the counselling and support to carry out the project work.

We are also thankful to **Dr. Chandramouli V S A**, **Principal**, BVC Engineering College, Odalarevu, for his support during and till the completion of the project.

We are thankful to all the **teaching and non-teaching staff** of Computer Science & Engineering Department, **Management**, BVC Engineering College, Odalarevu, and our friends for their direct and indirect help provided to us in completing the project.

Project Associates

**Sishir Bohara (18221A05B9)**

**Rupali Adhikari (18221A05C0)**

**Devram Yadav (18221A05B8)**

**FETCHING WEATHER INFORMATION**

**INTRODUCTION**

With the evolving Era of Technology, the society and world has come this far with the different Marvelous workshop Science and Technology. The use of technology can be in any field like : Education, Health, Transport, Banking and many more which we come across on our daily life. The mobile phone, Laptop, TV etc are the workshop science which contains very small or huge part of Computer Science.

In this Project, we will see one of the creation which is small to think but has a very important play in today’s time. The title of our project is “Fetching Weather Information” where we see the weather of different Cities in current time around the world.

Here we use several programming skills which consist of Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), JavaScript (JS). The platforms we have used in this project are Github, Virtual Studio, Netlify. The following are the topics to learned and look at first to continue with the project.

1. **HTML**

**The History of HTML**

* HTML was first created by Tim Berners-Lee, Robert Cailliau, and others starting in **1989**. It stands for Hyper Text Markup Language.
* Hypertext means that the document contains **links that allow the reader to jump to other places** in the document or to another document altogether. The latest version is known as [HTML5](https://html.com/html5/).
* A **Markup Language** is a way that computers speak to each other to control how text is processed and presented. To do this HTML uses two things: tags and **attributes.**

**Description:**

* HTML (Hyper Text Markup Language) is a language for specifying how text and graphics appear on a web page.
* When you visit a web site (e.g., www.google.com) your web browser retrieves the HTML web page and renders it.
* The HTML page is actually stored on the computer that is hosting the web site and the page is sent to your browser.
* HTML code is stored in a simple text file that has either a .htm or a .html filename extension (e.g., restaurant.html).
* A webpage is a document written in HTML and can be viewed on any web browser.
* It is contained within the web server, which can be accessed by entering the URL for that web page, and once it is loaded, it appears on the user's web browser.
* Each webpage is linked with a unique URL,hence two pages cannot have the same URL.
* Webpage is a part of a website; it means a website contains different web pages. Such as google.com is a website, and the page currently you are accessing is the webpage.
* The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.
* HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

**Syntax of HTML:**

<!DOCTYPE html>  
<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
  
<h1>Heading1</h1>

<h2>Heading2</h2>

<h3>Heading3</h3>

<h4>Heading4</h4>

<h5>Heading5</h5>

<h6>Heading6</h6>

<p>paragraph.</p>  
  
</body>  
</html>

**Output:**

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Paragraph

1. **CSS**

* CSS stands for Cascading Style Sheets.
* External stylesheets are stored in CSS files
* CSS is the language we use to style an HTML document.
* CSS describes how HTML elements should be displayed.
* This tutorial will teach you CSS from basic to advanced.
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
* CSS saves a lot of work. It can control the layout of multiple web pages all at once.
* External stylesheets are stored in CSS files.
* CSS removed the style formatting from the HTML page!

**Syntax:**



**Example:**

body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}

**There are three different ways you can use to insert CSS definitions in your web page. These are:**

1. **Inline Style**
2. **Embedded Style Sheet**
3. **External Style Sheet**

Inline CSS: Styles sheets that are of type inline refer to information related to the style being functional to the existing HTML element. Using an inline approach, rather than defining your style once, you have to write the style in every HTML element you use to design your web page. It can be more precisely called an inline style rather than the inline style sheet. It uses the style attribute within that HTML element.

Syntax:

<HTML ELEMENT style="properties: value"> .... </HTML ELEMENT>

Example:

<p style="color: blue">The text gets the effect of inline style.</p>

Embedded Style Sheet :Embedded Style Sheets is a style sheet where designers can embed information of the style sheet in an HTML document by making use of the <style> element. This embedding of style sheet info within <style> … </style> tags are done within head section of HTML.

The syntax for embedded style sheets has no such exception. Simply you have to place the style sheet code between the <head>.....</head> tags where <style> .... </style> is nested within head element:

Example:

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Embedded Style Sheets Example</title>

<style>

h1 {

border-bottom: 1px solid #DDDDDD;

color: #069;

font-family: Helvetica, Arial;

font-size: 25px;

font-weight: normal;

line-height: 34px;

margin-bottom: 10px;

outline: 0 none;

padding-bottom: 3px;

padding-top: 0;

text-decoration: none;

}

hr {

background-color: #069;

border: 0 none;

clear: both;

color: #D4D4D4;

height: 1px;

}

.sublines {

background-color: #DAF4FE;

padding: 5px;

border: 1px solid #09C;

font-family: Arial, Helvetica, sans-serif;

font-size: 13px;

}

.infotext {

font-size: 10pt;

background-color: #F2F2F2;

padding: 5px;

}

</style>

</head>

<body>

<h1><span class="headlines">Welcome to w3schools.in</span><br>

</h1>

<div class="sublines"> This is an example page using CSS.

<br> The example is really simple,

<br> and doesn't even look good,

<br> but it shows the technique. </div>

<br>

<table border="0" cellpadding="3" cellspacing="1">

<tr>

<td class="sublines"> As you can see:

<br>

</td>

<td class="sublines">The styles even work on tables.</td>

</tr>

</table>

<hr>

<div class="infotext">Example from w3schools.in</div>

<hr>

</body>

</html>

External Style Sheet : This type of style sheet gets a separate file in which designers can state every CSS styles that seem relevant for your web site.

Then this has to be linked with the external style sheet from your HTML page. You have to follow some specific steps to make this conceptual style sheet implementable.

Steps to create External Style Sheets:

1. Build the Style Sheet by typing the CSS code in a plain text file (using text editor, usually), and then save the with as a .css extension.
2. You have to link the Style Sheet with the HTML document by using an HTML link element.

Example:

<head>

<link rel="stylesheet" href="style.css" />

</head>

1. **JavaScript**

* JavaScript is the world's most popular programming language.
* JavaScript is the programming language of the Web.
* JavaScript is easy to learn.
* This tutorial will teach you JavaScript from basic to advanced.
* 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.
* JavaScript can be implemented using JavaScript statements that are placed within the <script> …. </script> HTML tags in a web page.
* You can place the <script> tags, containing your JavaScript, anywhere within you web page, but it is normally recommended that you should keep it within the <head> tags.
* The <script> tag alerts the browser program to start interpreting all the text between these tags as a script. A simple syntax of your JavaScript will appear as follows:

<script>

JavaScript code

</script>

The script tag takes two important attributes:

• Language: This attribute specifies what scripting language you are using. Typically, its value will be javascript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.

• Type: This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

So your JavaScript syntax will look as follows:

<script language=”javascript” type=”text/javascript”>

JavaScript code

</script>

Example:

<html>

<body>

<script language="javascript" type="text/javascript">

<!--

document.write ("Hello World!")

//-->

</script>

</body>

</html>

Output:

Hello World!

***API: (Application programming interfaces)***

* Application programming interfaces, or APIs, simplify software development and innovation by enabling applications to exchange data and functionality easily and securely.
* An application programming interface, or API, enables companies to open up their applications’ data and functionality to external third-party developers, business partners, and internal departments within their companies.
* This allows services and products to communicate with each other and leverage each other’s data and functionality through a documented interface.
* Developers don't need to know how an API is implemented;
* they simply use the interface to communicate with other products and services.
* API use has surged over the past decade, to the degree that many of the most popular web applications today would not be possible without APIs.
* An API is a set of defined rules that explain how computers or applications communicate with one another.
* APIs sit between an application and the web server, acting as an intermediary layer that processes data transfer between systems.

**Common API examples:**

* **Universal logins**
* **Third-party payment processing**
* **Travel booking comparisons**
* **Google Maps**

**Types of APIs**

* **Open APIs**
* **Partner APIs**
* **Internal APIs**
* **Composite APIs**