

Title of Project : Weather App

1.0 Aim /Benefits of the Micro-Project:

This Micro-Project aims at:

1. Build a Weather App that fetches weather data from an API by using HTML, JavaScript, CSS.
2. To show the weather forecast of an particular region.
3. To show current timing and date.

2.0 Course Outcomes Addressed

(Tick appropriate COs)

1. Create interactive web pages using program flow control structure [✓]
2. Implement Arrays and functions in javascript [✓]
3. Create event based web forms using javascript [✓]
4. Use javascript for handling cookies []
5. Create interactive webpage using regular expressions for validations.[]
6. Create menu and navigation in webpage []

3.0 Proposed Methodology

First Phase :- Data Gathering, rquirements, information collection.

Second Phase :- Analysis will be done to define functional requirements of the project.

Third Phase :- On the basis of analysis the next step will be writing an abstract.Abstract which gives brief introduction about the project.

Fourth Phase :- Step wise procedure writing which explain the working and the functionality of project.

Fifth Phase :- Code was tested and write the code and display the output.

Sixth Phase :- Project report was made

Seventh Phase :- Project was demonstrated and submitted.

1.0 Rationale

Importance :

JavaScript is limited featured client side programming language. JavaScript runs at the client end through the user's browser without sending message back and forth to the server. It is widely used by the developers to do things such as build dynamic webpages, respond to events, create interactive forms, validate data that the visitors enters into a form, control the browser etc. This course helps student to create highly interactive web pages using these features. Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser. It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content. The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field. The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server. JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

2.0 Aim of the Micro-Project:

This Micro-Project aims at:

1. Build a Weather App that fetches weather data from an API by using HTML, JavaScript, CSS.
2. To show the weather forecast of an particular region.
3. To show current timing and date.

3.0 Course Outcomes Addressed

1. Create interactive web pages using program flow control structure
2. Implement Arrays and functions in javascript
3. Create event based web forms using javascript

4.0 Literature Review

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as **LiveScript**, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name **LiveScript**. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript language.

- JavaScript is a lightweight, interpreted programming language.
- Designed for creating network-centric applications.
- Complementary to and integrated with Java.
- Complementary to and integrated with HTML.
- Open and cross-platform

Client-Side JavaScript :

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript:

The merits of using JavaScript are –

1. **Less server interaction** – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
2. **Immediate feedback to the visitors** – They don't have to wait for a page reload to see if they have forgotten to enter something.
3. **Increased interactivity** – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
4. **Richer interfaces** – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript:

We cannot treat JavaScript as a full-fledged programming language. It lacks the following important features –

1. Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
2. JavaScript cannot be used for networking applications because there is no such support available.
3. JavaScript doesn't have any multi-threading or multiprocessor capabilities.

Once again, JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

5.0 Actual Methodology Followed

Description :

We learn how to build a simple, yet fully functional weather app with JavaScript. First things first, we have to find a provider that will let us incorporate its weather data into our app. Luckily enough, there are several different providers out there for developing weather apps. Most of them include a free package along with premium subscriptions that scale depending on the services/features. In our case, we're going to use OpenWeatherMap, one of the most popular free choices. To take advantage of its capabilities, first, we have to sign up for an API key

The Page Markup:

The first section will include a heading, a search form, and an empty span element. This element will become visible with an appropriate message under certain conditions. Specifically, if there isn't any weather data available for a requested city or the data for this city are already known.

Some Basic Styles:

With the markup for the app ready, we'll forge on with the CSS. The first step, as always, is to specify some CSS variables and common reset styles:

Add the JavaScript :

On Form Submission

Each time a user submits the form by pressing the **Enter** key we'll do two things:

1. Stop the form from submitting, hence prevent reloading the page.
2. Grab the value which is contained in the search field.

CODING:

1. index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title> Weather App</title>
    <link rel="stylesheet" href="style1.css" />
    <script src="script.js" defer></script>
  </head>
  <body>

    <div class="column side">
      <p id="demo"></p>
      <p id="demo1"></p>
    </div>

    <div class="column side">
      <h1>Weather Forecast</h1>
    </div>

    <div class="abc">
      <form id="form">
        <input
          type="text"
          id="search"
          placeholder="Search by location"
          autocomplete="off" />
        <p id="write">City , Country</p>
      </form>
    </div>
  </body>
</html>
```

2. Style.css

```
* {
    box-sizing: border-box;
}

Body
{
    padding: 0px;
    margin: 0px;
    /*background-image: url("weather.jpg");*/
    background: linear-gradient( rgba(100, 77, 202, 0.37), rgba(231, 237, 238, 0.418) ),
url('weather.jpg');
    background-repeat: no-repeat;
    background-size: cover;
    background-position: top center;
    min-height: 100vh;
}
.column
{
    float: left;
    padding: 0px;
    margin: 0px;
}
.column.side
{width: 50%;
}
.abc
{
text-align: center;
}

#demo{
    padding: 5px;
    margin: 5px;
    font-size: 50px;
    font-weight: bold;
    font-family: Times New Roman;
    color: white;
}
```

```

#demo1{
    padding: 5px;
    margin: 5px;
    font-size: 40px;
    font-weight: bold;
    font-family: Comic Sans MS;
    color: white;
}
input[type=text]{
    background-color: #fff;
    border: none;
    border-radius: 30px;
    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
    font-family: Comic Sans MS;
    font-size: 1.5rem;
    padding: 1.5rem;
    min-width: 350px;
}
input:focus {
    outline: none;
}
#main{
    text-align: center;
    font-family: Comic Sans MS;
    font-size: 40px;
    font-weight: bold;
    color: black;
}
#write{
    margin: 0px;
    padding: 0px;
    font-size: 40px;
    color: black;
    font-weight:bold;
    font-family:Comic Sans MS;
}

```



```
h1{  
    padding: 0px;  
    font-size: 60px;  
    text-shadow: 6px 6px 8px red;  
    text-align: right;  
    padding-bottom: 30px;  
    text-align:center;  
    color: yellow;  
  
}
```

3. script.js

```
const apikey = "4d8fb5b93d4af21d66a2948710284366";
const main = document.getElementById("main");
const form = document.getElementById("form");
const search = document.getElementById("search");
Const url = (city) =>
`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apikey}`;
async function getWeatherByLocation(city)
{
    const resp = await fetch(url(city), { origin: "cors" });
    const respData = await resp.json();

    console.log(respData);

    addWeatherToPage(respData);
}

function addWeatherToPage(data)
{
    const temp = KtoC(data.main.temp);
    const weather = document.createElement("div");
    weather.classList.add("weather");
    weather.innerHTML = `
<h2>
${temp}°C
</h2>
<small>${data.weather[0].main}</small>`;
    // cleanup
    main.innerHTML = "";
    main.appendChild(weather);
}
function KtoC(K)
{
    return Math.floor(K - 273.15);
}
form.addEventListener("submit", (e) => {
    e.preventDefault();
    var a = document.getElementById("search").value;
    document.getElementById("write").innerHTML = a;
    const city = search.value;
```

```

    if (city)
    {
        getWeatherByLocation(city);
    }
});
var myVar = setInterval(myTimer, 0);
function myTimer() {
    var d = new Date();
    var t = d.toLocaleTimeString();
    document.getElementById("demo").innerHTML = t;
}
Myfunction();
function Myfunction(){
var day;
switch(new Date().getDay()){
case 0:
day="Sunday";
break;

case 1:
day="Monday";
break;

case 2:
day="Tuesday";
break;

case 3:
day="Wednesday";
break;

case 4:
day="Thursday";
break;

```

```
case 5:
day="Friday";
break;

case 6:
day="Saturday";
break;
}
var month;
switch(new Date().getMonth()){
case 0:
month="January";
break;

case 1:
month="February";
break;

case 2:
month="March";
break;

case 3:
month="April";
break;

case 4:
month="May";
break;

case 5:
month="June";
break;

case 6:
month="July";
break;
```

```
case 7:
month="August";
break;

case 8:
month="September";
break;

case 9:
month="October";
break;

case 10:
month="November";
break;

case 11:
month="December";
break;
}
var date=new Date().getDate()
document.getElementById("demo1").innerHTML = day+", "+month+" "+date;

}
```

6.0 Actual Resources Used

S. No.	Name of Resource/material	Specifications	Qty	Remarks
1	Hardware computer system	Computer (i3-i5 preferable),RAM minimum 2 GB and onwards but not limited to	1	I3 Processor
2	Operating system	Windows XP/windows 7/linux version 5 or later	1	Windows 7
3	Software	Notepad++	1	x86

7.0 OUTPUT





8.0 Skills developed out of this project.

1. Develop a JavaScript Program Using different Concepts
2. Able to debug and run a Javascript Program
3. Documentation of Project
4. Time management
5. Technical Writing
6. Team work
7. Problem solving

9.0 Applications of this Micro-Project

1. Websites
2. Web application
3. Information

Annexure - III

Suggested Rubric for Assessment of Micro-Project

S. No.	Characteristic to be assessed	Poor (Marks 1 - 3)	Average (Marks 4 - 5)	Good (Marks 6 - 8)	Excellent (Marks 9- 10)
1	Relevance to the course	Relate to very few LOs	Related to some LOs	Take care of at-least one CO	Take care of more than one CO
2	information collection	Not more than two sources very old reference	At-least 5 relevant sources, at least 2 latest	At –least 7 relevant sources, most latest	About 10 relevant sources, most latest
3	Completion of the Target as per project proposal	Completed less than 50%	Completed 50 to 60%	Completed 60 to 80%	Completed more than 80 %
4	Analysis of Data and representation	Data neither organized nor presented well	Sufficient and appropriate enough data generated but not organized and not presented well.	Sufficient and appropriate enough data generated which is organized and but not used.	Enough data collected and sufficient and presenting data.
5	Quality of Prototype/Model	Incomplete Programming code	Just assembled and some code is not functioning well.	Well assembled and functioning parts. But no creativity in design and use of graphics function	Well assembled with proper functioning parts.. Creativity in design and use of graphics function
6	Report Preparation	Very short, Details about methods, and conclusions omitted, some details are wrong	Nearly sufficient and correct details about methods, and conclusion. but clarity is not there in presentation.	Detailed, correct and clear description of methods and Conclusions. Sufficient Graphic Description.	Very detailed, correct, clear description of methods, and conclusions.
7	Presentation	Major information is not included, information is not well organized .	Includes major information but not well organized and not presented well	Includes major information and well organized but not presented well	Well organized, includes major information ,well presented
8	Defense	Could not reply to considerable number of question.	Replied to considerable number of questions but not very properly	Replied properly to considerable number of question.	Replied most of the questions properly

Micro Project Evaluation Sheet

Name of Student: Shadab Khan, Deepesh Kunkulol, Enrollment No: 1900040166, 1900040169,
 Jyoti Kolhe, Gaurav Thombare, 1900040170, 1900040172
 Pratham Shah 1900040175

Name of Program: Computer Engineering Semester: 5th

Course Title: Client Side Scripting

Course Code: 22519

Title of the Micro-Project: WHEATHER APP

Course Outcomes Addressed

(Tick appropriate COs)

1. Create interactive web pages using program flow control structure [✓]
2. Implement Arrays and functions in javascript [✓]
3. Create event based web forms using javascript [✓]
4. Use javascript for handling cookies []
5. Create interactive webpage using regular expressions for validations []
6. create menu and navigation in webpages []

Sr. No .	Characteristic to be assessed	Poor (Marks1-3)	Average (Marks 4-5)	Good (Marks 6-8)	Excellent (Marks9-10)	Sub Total
(A) Process and Product Assessment (Convert Above Total marks out of 6 Marks)						
1	Relevance to the course					
2	Information Collection					
3	Project Proposal					
4	Completion of the Target as per project proposal					
5	Analysis of Data and representation					
6	Quality of Prototype/Model					
7	Report Preparation					
(B) Individual Presentation / Viva (Convert above total marks out of 4 marks)						
8	Presentation					
9	Defense					

Roll No.	Process and Product Assessment (6 Marks)	Individual Presentation / Viva (4 Marks)	Total Marks 10
69			
72			
73			
75			
78			

Comments/Suggestions about team work/leadership/inter-personal communication

The project was implemented with good leadership & teamwork with active co-operation of all the team members. Everyone contributed equally and met with good outcome. Hence, overall project was done with harmony with good team spirit.

Name and designation of the Faculty Member: Mrs. Pratibha Pednekar

Signature _____