SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR

Activity-Based Learning Activity-1 & 2

(Website Design Activity)

on

ABL-1: Weather Website Using Bootstrap
ABL-2: Digital Clock using HTML, CSS and Javascript

Submitted in the partial fulfilment of the requirements for III Semester

Web Programming (S3CCSI04)

Submitted by

JAHNVI SHARMA	**1SI23CS116**
MRINALINI	**1SI23CS074**

Siddaganga Institute of Technology, Tumkur

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

B.H. road, Tumkur 572103, Karnataka, India

AY-2024-25

SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMAKURU - 3

CERTIFICATE

This is to certify that Activity Based Learning Avtivity 1 on "Web Site Design using BOOTSTRAP" and Activity 2 on "Web Site Design for Digital Clock" is a bonafide work carried out by **JAHNVI SHARMA (1SI23CS074) and MRINALINI (1SI23CS116)** of **III** semester Bachelor of Engineering in **Computer Science & Engineering** of the SIDDAGANGA INSTITUTE OF TECHNOLOGY during the academic year 2024-2025.

Signatures of Student	Faculty
i)	**Dr. Pramod T.C**
	Associate Professor
ii)	**Dept. of CSE,**
	SIT, Tumkur

TABLE OF CONTENTS ABL-1

ABL-2

INTRODUCTION

WEATHER WEBSITE (ABL 1):

Our *Weather App* is a simple and interactive web application that provides real-time weather information for any city. Built using *JavaScript, **Bootstrap, and **OpenWeatherMap API*, it allows users to easily check the current weather, including temperature, wind speed, humidity, and atmospheric pressure, by simply entering the name of a city.

Key Features:

1. *City Input*: Users can enter a city name, and the app fetches weather data for that location.
2. *Real-Time Weather Data*: Displays the current temperature in Celsius, weather description, wind speed, humidity, and pressure.
3. *Weather Icon*: Shows a corresponding weather icon that visually represents the weather condition.
4. *Current Date and Time: The app also shows the current date and time in a user-friendly format, using **Moment.js* for accurate formatting.
5. *Stylish UI: The app has a modern and attractive interface, utilizing **Bootstrap* for responsive design and *CSS animations* for smooth transitions.

This app is powered by the *OpenWeatherMap API* and is built to be both easy to use and visually engaging. Whether you're curious about the weather in your city or planning for travel, this app provides a seamless and interactive way to get up-to-date weather information.

DIGITAL CLOCK (ABL2):

This Digital Clock is a dynamic, interactive web-based clock that displays the current time in a 12-hour format (AM/PM), along with the current day of the week, date, and month. Built using JavaScript and styled with CSS, the clock updates every second to provide real-time accuracy.

Features:

Current Time: The time is displayed in a digital format with hours, minutes, and seconds. The clock automatically updates every second using setInterval().
AM/PM Format: The clock toggles between AM and PM to match the 12-hour time format.
Date and Day: It also shows the current day of the week (e.g., Monday, Tuesday) and the full date, including the month, day of the month, and year.
Responsive Layout: The clock is centered on the screen and designed to be visually appealing with gradient backgrounds and modern typography.

<u>Key Technologies Used:</u>

- JavaScript for handling the logic and dynamic updates of the time and date.
- CSS for styling the clock, creating gradients, and giving the display an appealing look.
- HTML to structure the content of the clock.
- This digital clock can be easily embedded into any webpage for use as a sleek, functional time display.

Weather Website (BOOTSTRAP (ABL 1))

```
<!DOCTYPE html>
<head>
<title>Weather App</title>
<link rel="stylesheet" href=
"https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"> <link
rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"> <link
rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.1/css/all.min.css">
<link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;700&di
splay=swap" rel="stylesheet">
</head>
</head>
<body class="bg-gradient">
<div class="container mt-5"> <div
class="card mx-auto text-center p-4
shadow-lg rounded-3
animate__animated animate__fadeInDown"
style="background: linear-gradient(to right, #f3a37e, #292e49); max-width:
500px;">
<h2 class="card-title mb-4 display-5 fw-bold" style="font-family:
'Montserrat',
sans-serif; color: rgb(46, 230, 61);">
Weather
App
</h2>
<div class="mb-3"> <label
for="city-input"
class="form-label visually-hidden">
Enter City
</label>
<div class="input-group"> <input
type="text" class="form-control form-
control-lg" id="city-input"
placeholder="Enter City">
<button class="btn btn-primary" onclick="getWeather()"
style="background-color: #FF6347; border-color: #FF6347;"> Get
Weather
</button>
</div>
</div>
<div id="weather-info"
```

```
class="mt-4 d-none animate__animated animate__fadeIn">
<h3 id="city-name"
class="mb-0 fs-7 fw-bold" style="color: #FFD700;"></h3>
<p id="date" class="text-muted mb-3 fs-6"></p>
<img id="weather-icon" class="mb-3"
alt="Weather Icon" style="width:
90px; height: 90px;">
<p id="temperature" class="mb-1 fs-
2 text-white fw-bold" style="color:
#FFD700;"></p>
<p               id="description"
class="mb-3   fs-4   text-white"
style="color: #FFD700;"></p>
<p id="wind-speed" class="fs-5 text-white"
style="color: #FFD700;"></p> <div
id="extra-info" class="mt-4">
</div>
</div>
</div>
</div>
<script src=
"https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js">
</script>
<script src=
"https://code.jquery.com/jquery-3.6.0.min.js">
</script>
<script src=
"https://momentjs.com/downloads/moment.min.js">
</script>
<script src="index.js"></script>
</body>

</html>
```

Javascript:

```
const API_URL = 'https://api.openweathermap.org/data/2.5/weather'; const
API_KEY = '97db128bda5b85f0638afbf11b2cd533';

function getWeather() {

const cityName = document.getElementById('city-input').value.trim() || 'Noida';

if (!cityName) {
alert('Please enter a city.');
return; }
Const
fullUrl= `${API_URL}?q=${cityName}&appid=${API_KEY}&units=metric`;


fetch(fullUrl)
.then(response => { if
(!response.ok) {
throw new Error('City not found. Please try again.');
}
return response.json();
})
.then(data => {

displayWeather(data);
})
.catch(error => {
console.error('Error fetching weather data:', error); alert(error.message);
});
}

function displayWeather(data) {
document.getElementById('weather-info').classList.remove('d-none');
document.getElementById('city-name').textContent        =        `Weather     in
${data.name}`;
document.getElementById('date').textContent = moment().format('MMMM Do
YYYY, h:mm:ss a');
document.getElementById('weather-icon').src                                  =
`https://openweathermap.org/img/wn/${data.weather[0].icon}.png`;
document.getElementById('temperature').textContent = `${data.main.temp}°C`;
document.getElementById('description').textContent =
data.weather[0].description; document.getElementById('wind-
speed').textContent = `Wind Speed:
${data.wind.speed} m/s`; document.getElementById('extra-info').innerHTML
= `
<p style="font-weight: bold; font-size: 18px; color: white;">Humidity:
${data.main.humidity}%</p>
<p style="font-weight: bold; font-size: 18px; color: white;">Pressure:
${data.main.pressure} hPa</p>
`;
}
```

PROJECT CODE DIGITAL CLOCK (ABL 2)

```
<!DOCTYPE html>
<html lang="en">
```

```html
<head>
<meta charset="utf-8">
<title>Digital Clock</title>
<link rel="stylesheet" href="style.css">
</head>

<body>
<div id="dayIntro">
<p id="dayName"></p>
</div>
<div class="container">
<div class="dispClock">
<div id="time"></div>
</div>
</div>
<script src="index.js"></script>
</body>

</html>
```

JavaScript:

```javascript
setInterval(currentTime, 1000);

function currentTime()
{ let time = new Date(); let
dayName=time.getDay(); let
month=time.getMonth(); let
year=time.getFullYear(); let
date=time.getDate(); let
hour = time.getHours(); let
min = time.getMinutes();
let sec = time.getSeconds();

var am_pm = "AM";
if(hour==12) am_pm
= "PM";
if (hour > 12) {
hour -= 12; am_pm
= "PM";
} if (hour == 0)
{ hour = 12;
am_pm = "AM";
}

hour = hour < 10 ? "0" + hour : hour; min
= min < 10 ? "0" + min : min;
sec = sec < 10 ? "0" + sec : sec;

let currentTime = hour + ":" + min + ":" + sec +" "+ am_pm;

var
months=["January","February","March","April","May","June","July","August","S
eptember","October","November","December"]; var
week=["Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturd
ay"];

var presentDay=week[dayName]+", "+months[month]+" "+date+", "+year;

const clock = document.getElementById("time");
const dayIntro=document.getElementById("dayName");

clock.innerHTML = currentTime; dayIntro.innerHTML
= presentDay;
}

currentTime();
```

CSS:

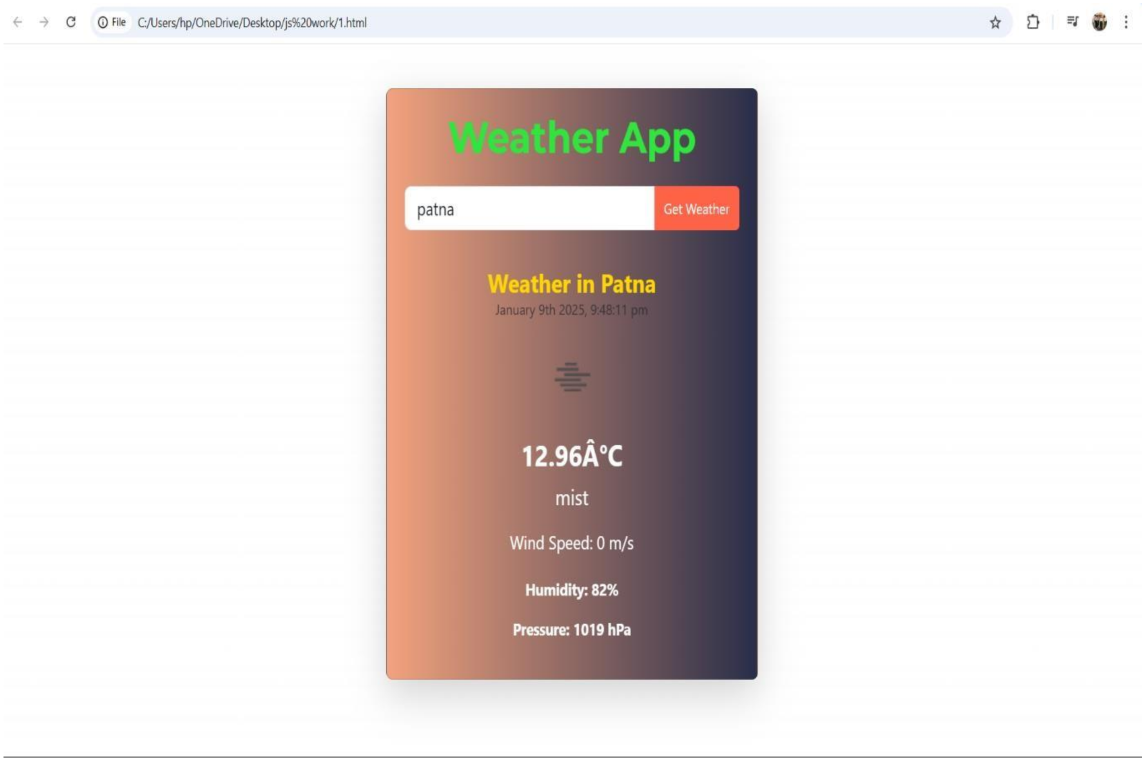
```
*{ margin:
0;
padding: 0;
```

```
} html,body{
display: grid;
place-items: center;

}
#dayIntro { font-size: 40px; font-
weight: 600; letter-spacing: 3px;
border: 7px solid rgb(17,129,134);
border-radius: 10px; margin:
20px;
font-family: 'Times New Roman', Times, serif; padding:
15px;
background: linear-gradient(180deg, #a8b9d3,rgb(173, 227, 229));
} .container{
height: 120px;
width: 550px;
position: relative;
background: linear-gradient(135deg, #14ffe9, #ffeb3b, #ff00e0); border-radius:
10px;
cursor: default;

}
.container .dispClock,
.container { position:
absolute; top: 28%;
left: 50%;
transform: translate(-50%, -50%);
}
.container .dispClock{
top: 50%; height:
105px; width: 535px;
background: linear-
gradient(147deg,
#000000 0%, #2c3e50
74%); border-radius:
6px; text-align:
center;
}
.dispClock #time{ line-height:
85px; color: #fff; font-size: 70px;
font-weight: 600; letter-spacing:
```

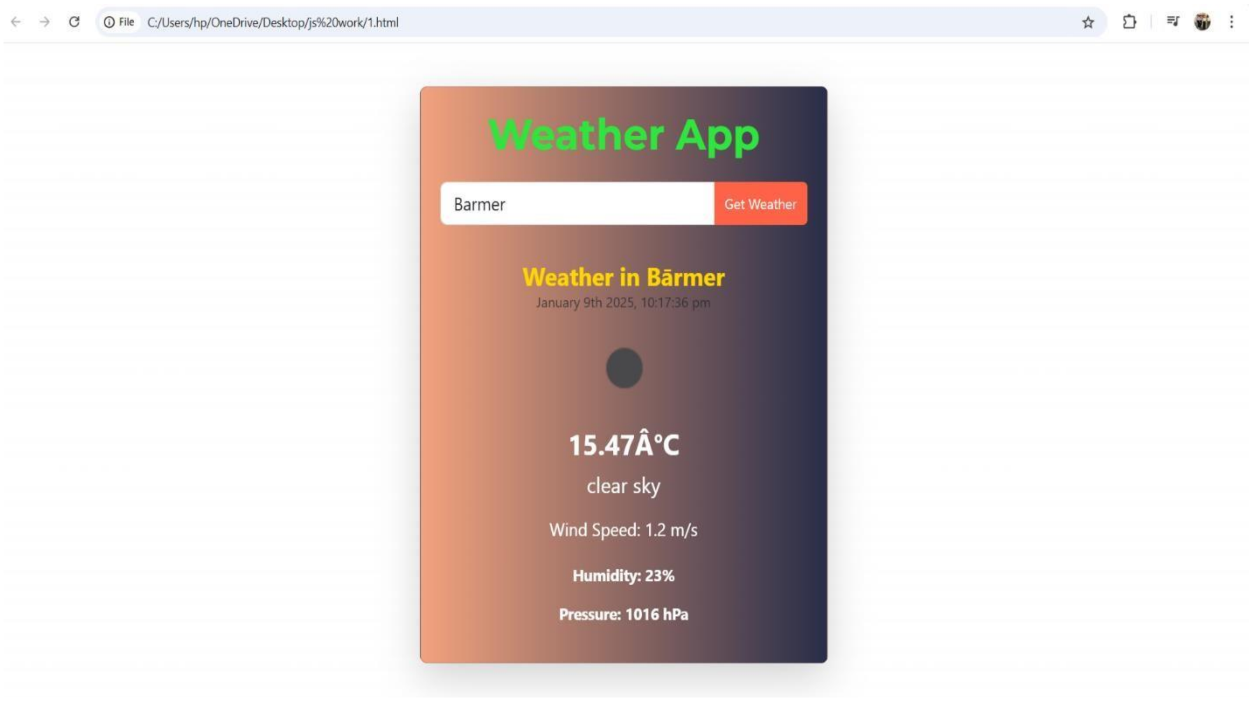
```
1px; font-family: 'Orbitron', sans-
serif;
background: linear-gradient(135deg, #14ffe9, #ffeb3b, #ff00e0);
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;
}
```

PROJECT OUTPUT (SCREENSHOTS)

ABL 1

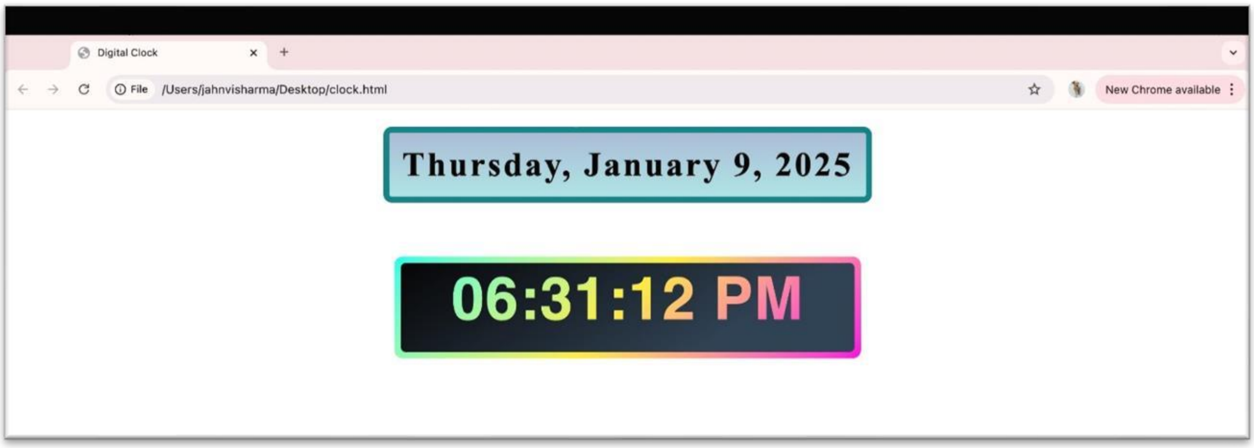


Weather in Patna
January 9th 2025, 9:48:11 pm
12.96Â°C
mist
Wind Speed: 0 m/s
Humidity: 82%
Pressure: 1019 hPa



Weather in Bārmer
January 9th 2025, 10:17:36 pm
15.47Â°C
clear sky
Wind Speed: 1.2 m/s
Humidity: 23%
Pressure: 1016 hPa

ABL 2


Digital Clock
File /Users/jahnvisharma/Desktop/clock.html
New Chrome available
Thursday, January 9, 2025
06:31:12 PM

LIST OF HTML TAGS USED AND CSS PROPERTIES

Project 1:

<u>HTML:</u>

- Document Structure:
  The document begins with <!DOCTYPE html>, indicating an HTML5 document.

The <html> element, which encompasses all content, is used (though not explicitly opened here).

The <head> tag includes metadata, external stylesheets (Bootstrap, animations, Font Awesome, Google Fonts), and the <title> tag to set the browser title.

- Content Section:

  The <body> holds the visual content, primarily structured with <div> elements.

  The main container (<div class="container">) wraps the entire content. Another <div class="weather-card"> defines the specific card structure for displaying weather details.
- Text & Labels:

  <h3> tags are used for titles, including the "Weather App" and dynamic city name.

  <p> tags display various weather-related data, such as date, temperature, description, and wind speed.
- User Interaction:

  An <input> field allows users to type in the city name.

  The <button> triggers the weather fetch when clicked, invoking the JavaScript function.
- Scripts:

  <script> tags link to external JavaScript files, including libraries like jQuery, moment.js for date formatting, and a custom script (index.js) that contains logic for weather fetching.

CSS:

- General Styles: • body:
  - Utilizes a gradient background transitioning from green (#4CAF50) to blue (#2196F3).
  - Uses flexbox for centering content both horizontally and vertically, with a height of 100% of the viewport (height: 100vh). o Font is set to Montserrat, with no default margin for clean positioning.
- Container and Card:

  container: o Applies center-aligned text using text-align: center. weather-card: o The card has a semi-transparent white background, rounded corners, and a soft shadow effect.
  - When hovered, it scales up slightly (using transform: scale(1.05)) for a smooth interactive effect.
- Form and Input Elements:

  #city-input: o The city input field is styled with padding and a border. When focused, it changes color.

    - The placeholder text is given a subtle light grey color.

  #city-input-btn: ○ The button styling includes padding, a blue background, white text, rounded corners, and no border. On hover, the background darkens.
- Weather Information:

  #weather-info: ○ Initially hidden (display: none;), it's made visible once the weather data is available.

#weather-icon:

  - The icon for weather condition is set to a fixed size of 100x100 pixels.

  #temperature, #description, #wind-speed, and #date:

  - #temperature is larger, with bold font and margin for separation.
  - #description is moderately sized, and #wind-speed is styled in red to emphasize the wind speed.

Project 2:

HTML:

Structural Tags:

<!DOCTYPE html>: Specifies the document type (HTML5).
<html>: The root element of the HTML document.
<head>: Contains metadata and links to external resources. <body>:
Contains the visible content of the webpage.

Metadata/Resource Tags:

<meta>: Defines metadata like character encoding (utf-8).
<title>: Sets the title of the webpage (visible in the browser tab).
<link>: Links an external CSS file (style.css) for styling.
<script>: Links an external JavaScript file (index.js) for functionality.

Content Tags:

<div>: Groups and organizes content. id="dayIntro":
Container for the day's name. id="time": Displays
the digital clock time. class="container": General
container for layout. class="dispClock": Container
for the clock display.
<p>: Paragraph element to display the day's name (inside id="dayIntro").

CSS:

- Layout and Positioning:

  display: grid;: Makes the html and body elements a grid container to center the content.
  place-items: center;: Centers content both horizontally and vertically within the grid. position: relative;: Used to position .container relative to its normal position. position: absolute;: Positions .dispClock and other elements relative to their closest positioned ancestor. top, left, transform: translate(-50%, -50%);: Used to center .container and .dispClock elements exactly in the middle.

- Font and Text Styling:

  font-size, font-weight, letter-spacing: Controls the font size, weight, and letter spacing for readability and design.
  font-family: Defines the typeface (e.g., 'Times New Roman', 'Orbitron') used in elements.
  line-height: Ensures proper vertical spacing within the text (specifically for the clock time).
  -webkit-background-clip: text;: Creates a text gradient effect by applying the background gradient to the text itself.

- Spacing and Alignment:

  margin, padding: Used for spacing around and inside elements. text-align: center;: Centers text within its container.

- Background and Colors:

  background: Specifies gradient backgrounds for elements like the day intro (linear-gradient) and the clock (background: linear-gradient with multiple colors).
  color: Specifies the color of text (used for white text in the clock and other sections).

- Borders and Box Model:

  border: Defines borders (e.g., for the day intro #dayIntro) with a specific width, style, and color.
  border-radius: Rounds the corners of containers and elements for a smooth look.

- Cursor:

cursor: default;: Changes the cursor to the default arrow when hovering over the .container element.