Mini Project Report

on

Weather Application

Submitted to

Ajay Kumar Garg Engineering College, Ghaziabad

(BTech Information Technology Sem 5th , 2020-21)

(KSC-554 Project Mini Project or Internship Assessment Report)

Submitted To: -	**Submitted By:-**
Dr. Ruchi Gupta	Shivam Gupta 1900270130157
Dr. Rashmi Sharma	Shivam Singh 1900270130159

Dr. A.P.J. Abdul Kalam Technical University,
Uttar Pradesh, Lucknow

Acknowledgement

This project is the work of several days study on various aspects of HTML, CSS, and JAVA SCRIPT Project. During the work, we have received a great amount of help from our Director General "**Dr. R.K. Agarwal**", of **AJAY KUMAR GARG ENGINEERING COLLEGE**, whom we wish to acknowledge and thank from the depth of our hearts.

We are grateful to "**Dr. Anu Chaudhary**" our HOD of IT Department, for giving us this golden opportunity to learn and showcase our skills through project work.

We are very much obliged and thankful to "**Dr. Ruchi Gupta**" and "**Dr. Rashmi Sharma**", our Coordinators for providing this opportunity and constant guidance given by him during the course.

At last, but not the least, we would like to show gratitude to all our friends for their support and guidance.

THANKING YOU

Shivam Gupta (1900270130157)

Shivam Singh (1900270130159)

ABSTRACT

Millions of people in the India regularly acquire essential information from weather forecasts for a wide variety of reasons. A myriad of sources exists for obtaining daily weather information, and the rapid growth in mobile device technology has created a very convenient means for people to retrieve this data. Languages used in developing a website works in stages by completing all the needs on every level. Html provides a skeleton while CSS stylesthe application and makes it more user friendly. However, that is notenough to make it interactive, therefore we use JavaScript.

Technology Used

Languages:-

1. Html
2. CSS
3. JavaScript

Softwares:-

Vs Code

Chrome

TABLE OF CONTENT

INTRODUCTION

In today day to day life everyone is busy in their work. No one have time to waste Everybody want to do their work fast without any error and want to complete their work on time so no one want to waste their single second. But some time due to unexpected things the problem is created. There are many unexpected things which are not in our control but with the help of technology we are able to avoid these unexpected things so thad people can have prior knowledge about that and they can manage their time according to that.

Now a days weather condition frequently changes due to which it become difficult for people to plan their task/daily activities as per their time table so With weather applications people can exactly know when to expect a weather change and take decisions accordingly.

Problem statement and description

Problem Statement:- In areas where weather frequently changes in a matter of minutes, it becomes difficult for people to plan their daily activities.

Reason:- With weather applications people can exactly know when to expect a weather change and take decisions accordingly.

Objective:- To make an application to aware users about weather by using approximate data from open weather map.

Methodology:-

The regional data about the weather conditions are being kept over open weathermap.

Step 1:- Take user input about the location of the region of which the weather condition needs to be displayed to the user.

Step 2:- Send the location as an input to the open weather map API to find the corresponding data.

Step 3:- Next is to display the required data on the users screen.

Step 4:- Alternatively the latitude and longitude of the location can also be used to fetch the weather conditions.

Contribution:-

- Fetch is easy to understand and widely used while calling API to get the data.
- The name of the region or the longitude or latitude can be used to fetch data and display.
- Open weather map is widely used to provide weather condition status to the users.

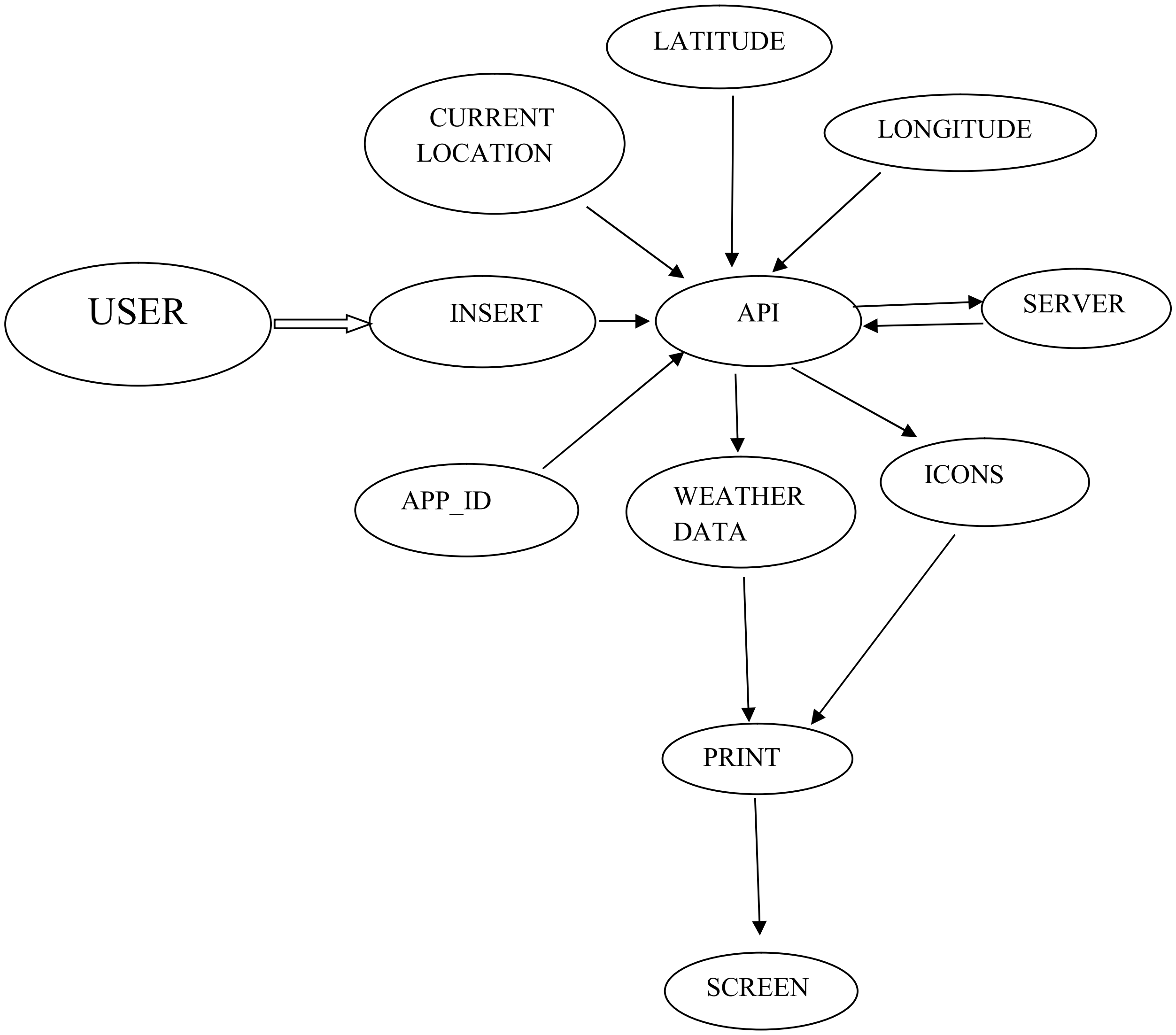
Limitations:-

- The data provided is approximate and may not match the actual weather conditions of the region.
- The user may also get the data of another region having same name when using limited information.
- The application will work in the presence of the internet connectivity only.
- The UI of the application might be unsuitable for the users.

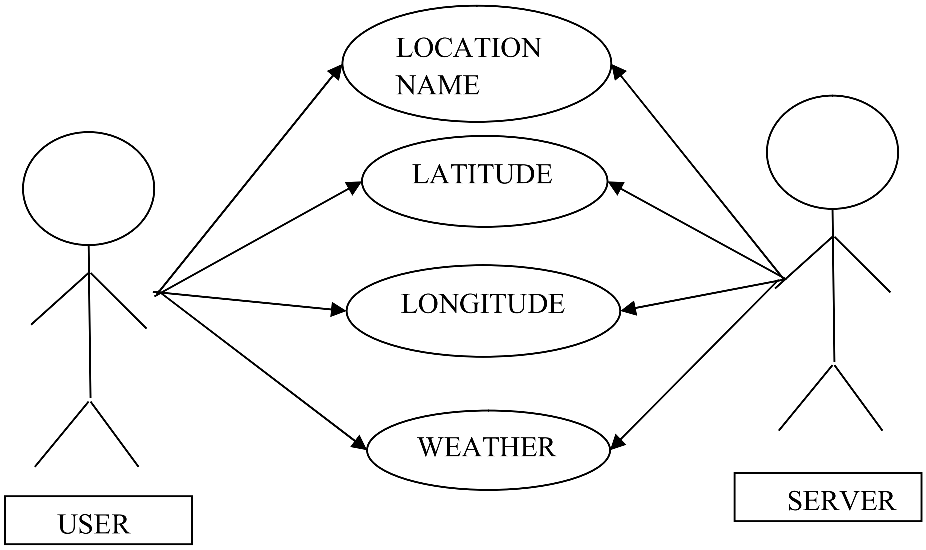
Hardware & Software:-

- AMD Ryzen 5 4600H
- 8GB RAM
- 500GB SSD
- IDE: Visual Studio Code
- Web Browser: Google Chrome

DATA FLOW DIAGRAM



SRS WITH USE CASE DIAGRAM



CODE

❖ HTML CODE:-

- **Index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<meta name="description" content="">
<title>Weather Application</title>
<link rel="stylesheet" href="index.min.css">

</head>

<body class='body'>

<div class='content'>
```

```
<header class='header'>
<div class='menu menu__btn menu--loc'></div>
<h1 class='h1'>Current weather for your location</h1>
</header>

<div class='row col992px'>
<div class='temp'></div>
<div class='row row__city'>
<div class='name'></div>
<div class='icon'></div>
</div>
</div>
<main class='weather row row__weather'>
<div class='hum'></div>
<div class='press'></div>
<div class='wind'></div>
</main>

<form onsubmit='return false' class='menu row row__menu'>
<input type='text' class='menu menu__city' placeholder='Different city
name'></input>
<div type='button' class='menu menu__btn menu--search'>Search</div>
</form>
```

```
</div>



<script src="index.js"></script>
</body>
</html>
```

❖ JAVA SCRIPT CODE:-

- **Index.js**

```
const APP_ID = '927d680a60df926004f14faf7f2773dc'

const searchButton = document.querySelector('.menu--search')
const locationButton = document.querySelector('.menu--loc')
const input = document.querySelector('.menu__city')
const city = document.querySelector('.name')
const icon = document.querySelector('.icon')
const temp = document.querySelector('.temp')
const hum = document.querySelector('.hum')
const press = document.querySelector('.press')
const windSpeed = document.querySelector('.wind')
const h1 = document.querySelector('.h1')

const getWeathetForLocation = () => {
const searchForCurrentLacotion = options => {
return new Promise((resolve, reject) => {
navigator.geolocation.getCurrentPosition(resolve, reject, options)
})
}
```

```
searchForCurrentLacotion()
.then((position) => {
const url = `https://api.openweathermap.org/data/2.5/weather?lat=${
position.coords.latitude }&lon=${ position.coords.longitude
}&units=metric&appid=${ APP_ID }`
h1.innerHTML = `Current weather for your location`
locationButton.style.display = `none`
getWeathet(url)
})
.catch((err) => {
console.error(err.message)
alert(err.message)
})
}

getWeathetForLocation()
locationButton.addEventListener('click', getWeathetForLocation)

searchButton.addEventListener('click', e => {
const cityName = input.value
const url = `https://api.openweathermap.org/data/2.5/weather?q=${
cityName }&units=metric&appid=${ APP_ID }`
```

```
h1.innerHTML = 'Current weather'
locationButton.innerHTML = `Get weather for your location`
locationButton.style.display = `block`
getWeathet(url)
})

input.addEventListener('keyup', e => {
e.keyCode === 13 && searchButton.click()
})

const getWeathet = url => (
fetch(url).then(response => {
if (response.status !== 200) {
console.error(`Status Code: ${ response.status }`)
}
return response.json()
}).then(data => {
const { sys, weather, main, wind } = data
city.innerHTML = `${ data.name }, ${ sys.country }`
icon.innerHTML = `<img src= 'https://openweathermap.org/img/w/${
weather[0].icon }.png'>`
temp.innerHTML = `${ Math.round(main.temp) }&deg;C`
hum.innerHTML = `Humidity: ${ main.humidity }%`
press.innerHTML = `Pressure: ${ main.pressure }hPa`
windSpeed.innerHTML = `Wind speed: ${ wind.speed }m/s`
}))
```

❖ SASS CODE:-

- **Index.scss**

```
@import
url(https://fonts.googleapis.com/css?family=Source+Sans+Pro);

.body, html{
margin: 0;
padding: 0;
width: 100%;
height: 100%;
min-height: 100%;
text-align: center;
background: linear-gradient(to right, #b3dced, #29b8e5);
font-family: 'Source Sans Pro';
color: #fff;
display: flex;
flex-direction: column;
}
```

```
.content{
flex: 1;
}

.header{
display: flex;
}

.h1{
font-size: 40px;
text-transform: uppercase;
color: #326292;
margin: 20px auto;
}

.name{
font-size: 40px;
letter-spacing: 0.3em;
margin-top: 60px;
}

.icon{
margin-right: 20px;
```

```
}

.row{
display: flex;
&__menu{
justify-content: flex-end;
}
&__weather{
justify-content: center;
}
}

.temp {
font-size: 160px;
flex: 0.9;
color: #326292;
}

.weather{
font-size: 28px;
display: flex;
margin: 0 40px;
& div {
```

```
padding: 20px;
}
}

.menu{
padding: 20px;
border-radius: 15px;
margin-left: 15px;
margin-bottom: 20px;
&__city{
width: 250px;
border: 4px solid #326292;
box-sizing: border-box;
}
&__btn{
cursor: pointer;
background-color: orange;
height: 25px;
}
&__btn:hover{
background-color: #326292;
}
&--loc{
```

```
width: 200px;
display: none;
margin: 20px;
}
}

.footer{
background-color: #326292;
padding: 10px 0;
flex-shrink: 0;
}

@media (max-width: 992px) {
.col992px{
flex-direction: column;
}
.row__city{
justify-content: center;
}
.temp{
order: 1;
}
.row__menu{
```

```
justify-content: center;
}
}

@media (max-width: 760px) {
.header{
flex-direction: column;
}
.h1{
font-size: 30px;
margin: 10px 0 15px;
}
.name{
font-size: 30px;
margin: 0;
letter-spacing: 3px;
}
.icon{
margin: -8px 0 0 10px;
}
.temp{
font-size: 100px;
}
```

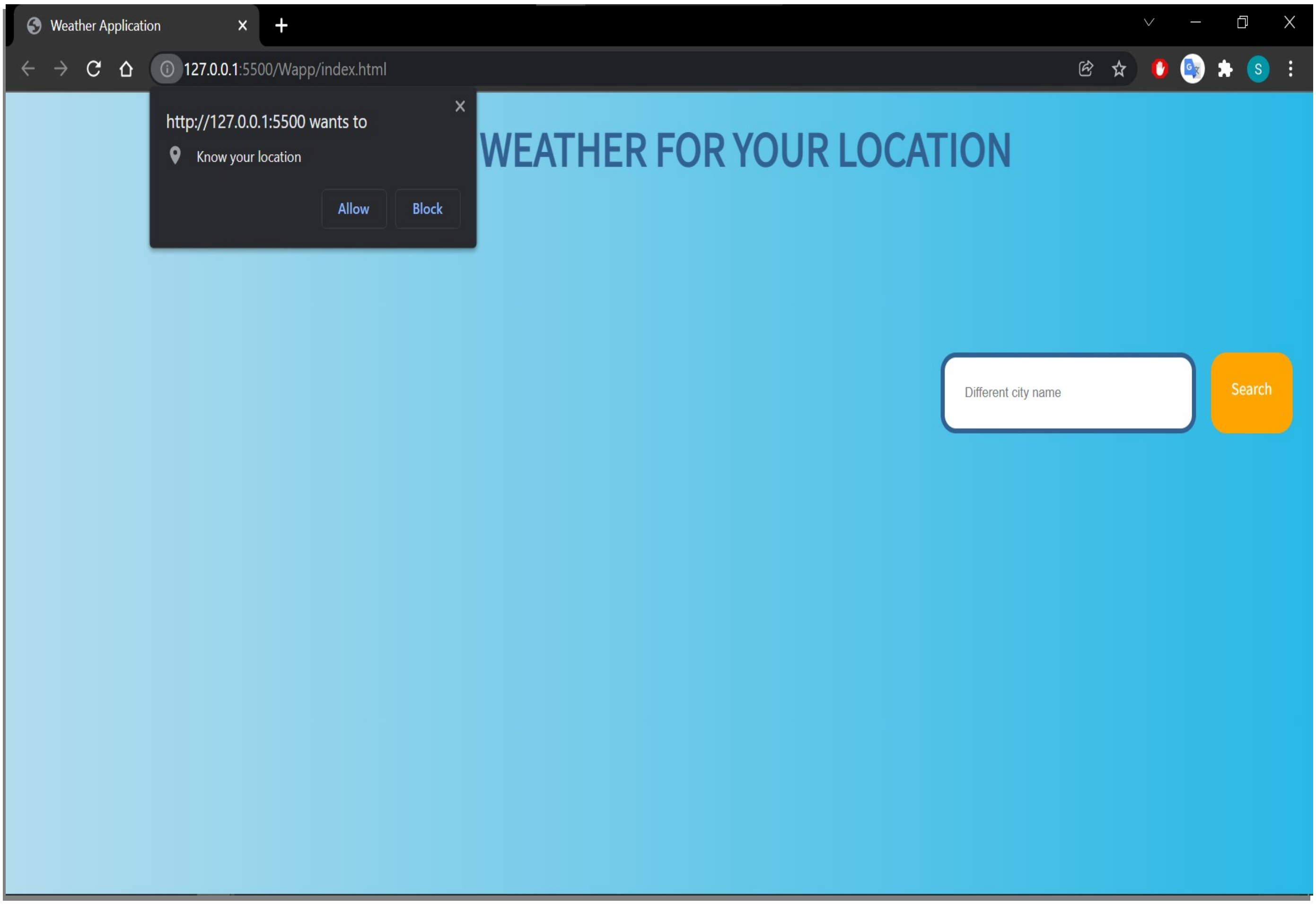
```
.menu--loc{
justify-content: center;
}
}

@media (max-width: 576px) {
.row{
flex-direction: column;
&__menu{
justify-content: center;
}
}
.menu{
margin: 12px auto;
&--search{
width: 210px;
}
}
}
```

❖ CSS CODE:-

- **index.min.css**

```
@import
url(https://fonts.googleapis.com/css?family=Source+Sans+Pro);.body,ht
ml{margin:0;padding:0;width:100%;height:100%;min-
height:100%;text-align:center;background:linear-gradient(to right,
#b3dced, #29b8e5);font-family:'Source Sans
Pro';color:#fff;display:flex;flex-
direction:column}.content{flex:1}.header{display:flex}.h1{font-
size:40px;text-transform:uppercase;color:#326292;margin:20px
auto}.name{font-size:40px;letter-spacing:0.3em;margin-
top:60px}.icon{margin-
right:20px}.row{display:flex}.row__menu{justify-content:flex-
end}.row__weather{justify-content:center}.temp{font-
size:160px;flex:0.9;color:#326292}.weather{font-
size:28px;display:flex;margin:0 40px}.weather
div{padding:20px}.menu{padding:20px;border-radius:15px;margin-
left:15px;margin-bottom:20px}.menu__city{width:250px;border:4px
solid #326292;box-sizing:border-
box}.menu__btn{cursor:pointer;background-
color:orange;height:25px}.menu__btn:hover{background-
color:#326292}.menu--loc{width:200px;display:none;m
argin:20px}.footer{background-color:#326292;padding:10px 0;flex-
shrink:0}@media (max-width: 992px){.col992px{flex-
direction:column}.row__city{justify-
content:center}.temp{order:1}.row__menu{justify-
content:center}}@media (max-width: 760px){.header{flex-
direction:column}.h1{font-size:30px;margin:10px 0 15px}.name{font-
size:30px;margin:0;letter-spacing:3px}.icon{margin:-8px 0 0
10px}.temp{font-size:100px}.menu--loc{justify-
content:center}}@media (max-width: 576px){.row{flex-
direction:column}.row__menu{justify-
content:center}.menu{margin:12px auto}.menu--search{width:210px}}
```

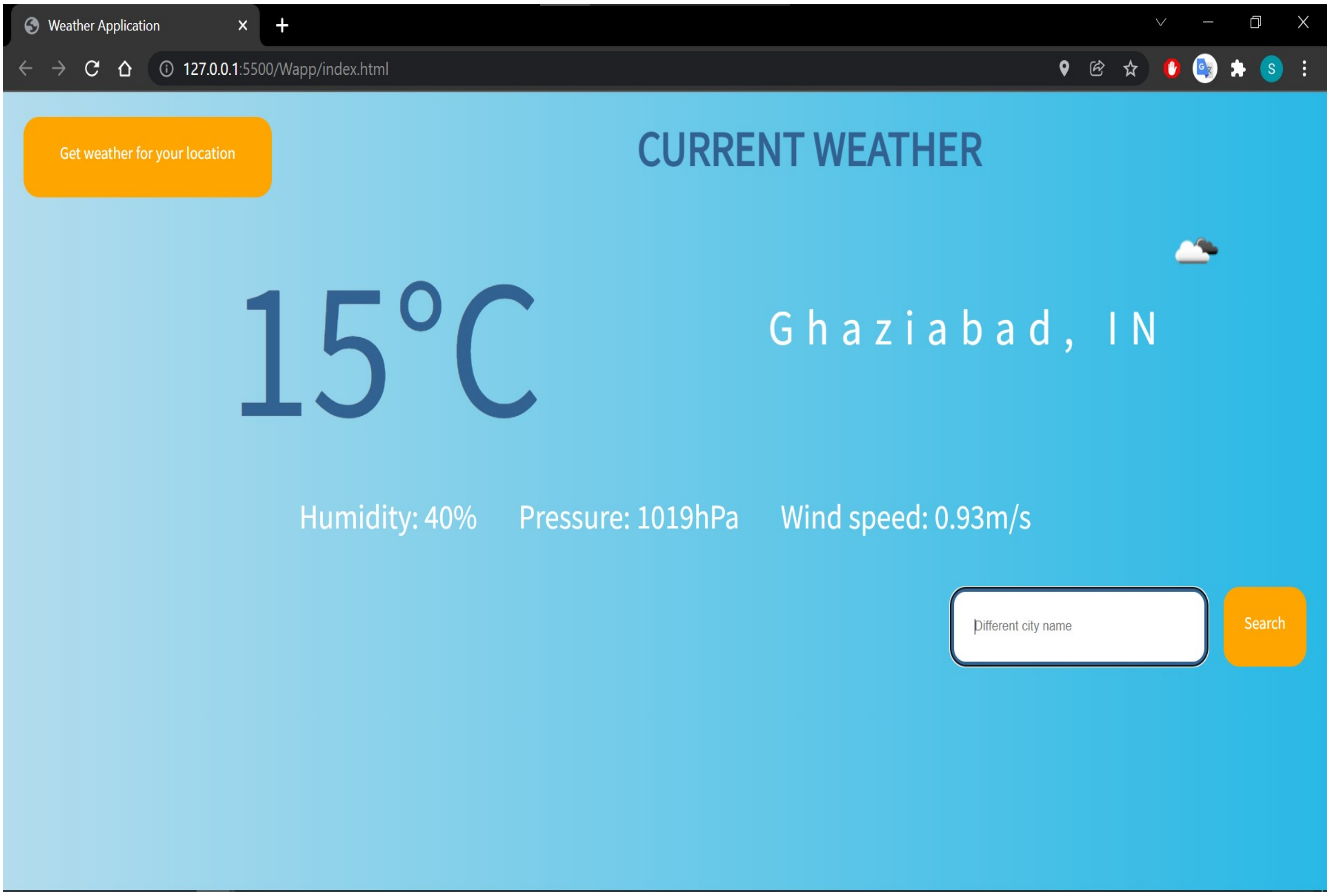
TEST RESULT



http://127.0.0.1:5500 wants to

Know your location

WEATHER FOR YOUR LOCATION

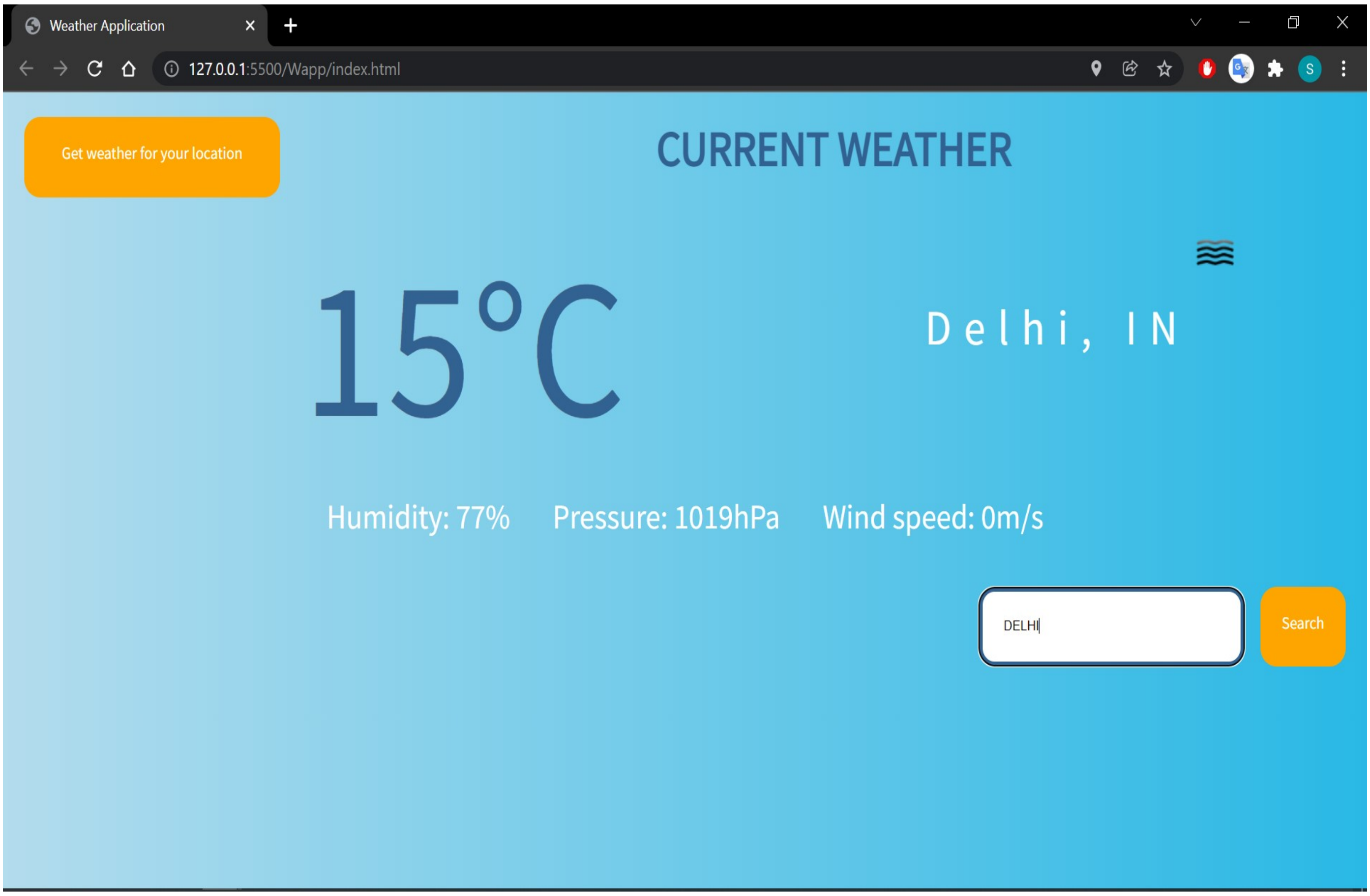


CURRENT WEATHER

15°C

Ghaziabad, IN

Humidity: 40%   Pressure: 1019hPa   Wind speed: 0.93m/s



Weather Application
127.0.0.1:5500/Wapp/index.html
Get weather for your location
CURRENT WEATHER
15°C
Delhi, IN
Humidity: 77%
Pressure: 1019hPa
Wind speed: 0m/s
DELHI
Search