

Front End Engineering-II

Project Report

Semester-IV (Batch-2022)

Weather App
(Using Bootstrap)



Supervised By:

Raveesh Samkaria

Submitted By:

Uday Kalra

2210990918

G-14

**Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab**

ABSTRACT

This project introduces the development of a dynamic "Weather App" application, designed to enhance user experience and efficiency in data browsing. Leveraging a combination of HTML, CSS, JavaScript, and Bootstrap, alongside integration with an external API for data retrieval, the application offers real-time filtering capabilities. The core objective is to empower users to swiftly navigate through a dataset by dynamically filtering search results as they type.

The project focuses on implementing intuitive and responsive search functionalities, enabling seamless interaction with the dataset. Through the utilization of modern web technologies and design principles, the application prioritizes user-centricity, aiming to streamline the information retrieval process. Key features include a visually appealing and intuitive user interface, responsive design for compatibility across devices, and efficient data fetching through API integration. The implementation of live search filtering aims to minimize user effort and enhance accessibility to relevant information.

This abstract provides a concise overview of the project's scope, highlighting the utilization of HTML, CSS, JavaScript, Bootstrap, and API integration to develop a "Live Search Filter" application. The project underscores the importance of user-centric design and efficiency in data browsing, contributing to a more seamless web browsing experience.

1. Introduction

In an era characterized by the exponential growth of digital information, efficient data retrieval mechanisms have become paramount. With the proliferation of online platforms and repositories, users often find themselves inundated with vast amounts of data, necessitating tools that enable streamlined access to relevant information. In response to this challenge, the "Live Search Filter" application emerges as a solution aimed at enhancing user experience and efficiency in navigating datasets.

1.1 Background:

The exponential expansion of digital data has necessitated the development of innovative tools to facilitate information retrieval. Traditional search methods often require users to sift through extensive datasets manually, resulting in time inefficiencies and diminished user satisfaction. Recognizing this challenge, the "Live Search Filter" application leverages modern web technologies to provide users with a dynamic and intuitive means of accessing information.

1.2 Objectives:

The primary objective of the "Live Search Filter" project is to develop an application that enables real-time search filtering on a dataset, thereby empowering users to efficiently navigate through information. By implementing dynamic search functionalities using HTML, CSS, JavaScript, and Bootstrap, alongside integration with an external API for data fetching, the project aims to create a seamless and interactive user experience. The application seeks to minimize user effort by dynamically filtering search results as users type, fostering a more efficient and user-centric approach to data browsing.

1.3 Significance:

The significance of the "Live Search Filter" application lies in its ability to address the evolving needs of users in the digital age. By providing a responsive and intuitive platform for

data retrieval, the application enhances user productivity and satisfaction. Additionally, the project contributes to the advancement of web development practices by showcasing the integration of modern technologies and design principles to optimize user experience. Furthermore, the application's adaptability across devices underscores its relevance in an increasingly mobile-oriented landscape, ensuring accessibility and usability for a diverse user base. Through its focus on efficiency, usability, and accessibility, the "Live Search Filter" application exemplifies the ongoing efforts to enhance the digital browsing experience.

Problem Definition:

The contemporary digital landscape is characterized by an abundance of data across various online platforms and repositories. Users often encounter challenges in efficiently navigating through these datasets to access relevant information. Traditional search mechanisms require users to manually sift through extensive data, resulting in time inefficiencies and diminished user satisfaction.

The problem at hand is the need for a solution that streamlines the process of data retrieval, offering users a more efficient and intuitive means of accessing information. Specifically, there is a demand for a dynamic search filtering mechanism that enables real-time refinement of search results as users interact with the search interface. This entails the development of a responsive and user-centric application capable of seamlessly integrating with external data sources through API integration.

The "Live Search Filter" project aims to address this problem by leveraging modern web technologies to create a dynamic and interactive search experience. By implementing real-time search filtering functionalities, the project seeks to enhance user productivity and satisfaction, ultimately providing a solution to the challenges associated with data browsing in the digital age.

Requirements:

The "Weather App" project aims to develop a dynamic web application that enhances user experience and efficiency in navigating datasets. Leveraging modern web technologies such as HTML, CSS, JavaScript, and Bootstrap, alongside integration with an external API for data fetching, the application offers real-time search filtering capabilities. Users can enter search queries and receive instant, dynamically filtered results, minimizing the time and effort required to find relevant information.

1. Search Functionality: The application should allow users to enter search queries and receive real-time filtered results.

2. Dynamic Filtering: As users type their search queries, the application should dynamically filter the dataset to display relevant results instantly.
3. Responsive Design: The application interface must be responsive, ensuring compatibility across various devices and screen sizes.
4. API Integration: Integration with an external API for data retrieval is essential to fetch the dataset dynamically.
5. Clear User Interface: The user interface should be intuitive and visually appealing, providing clear feedback on search results and interactions.
6. Search Functionality: The application should allow users to enter search queries and receive real-time filtered results.
7. Dynamic Filtering: As users type their search queries, the application should dynamically filter the dataset to display relevant results instantly.
8. Responsive Design: The application interface must be responsive, ensuring compatibility across various devices and screen sizes.
9. API Integration: Integration with an external API for data retrieval is essential to fetch the dataset dynamically.
10. Clear User Interface: The user interface should be intuitive and visually appealing, providing clear feedback on search results and interactions.

Proposed Design and Methodology:

Proposed design and methodology embrace an innovative approach that seamlessly integrates visual appeal with functional efficiency. This section outlines the envisioned design principles and the systematic methodology that guides the development process.

1.2 Design Concept:

The design of the "Weather App" application encompasses both the user interface and the underlying architecture. The user interface design focuses on providing a visually appealing and intuitive search experience, while the architectural design ensures efficient data retrieval and processing.

1.2 User-Interface Design:

The user interface design follows a clean and minimalist approach, with emphasis on simplicity and ease of use. Key components include:

1. Search Input Field: A prominent input field where users can enter search queries.
2. Search Results Display: Dynamically updated area to display filtered search results.
3. Visual Feedback: Clear visual cues to indicate search activity and results.
4. Search Input Field: A prominent input field where users can enter search queries.
5. Search Results Display: Dynamically updated area to display filtered search results.
6. Visual Feedback: Clear visual cues to indicate search activity and results.
7. External API : Integrated to fetch the dataset dynamically based on user search queries.

By adopting this design and methodology, the "Live Search Filter" project aims to deliver a robust and user-friendly application that effectively meets the needs of its target audience.

This proposed design and methodology encapsulate our commitment to creating a cutting-edge Live Search Filter. By harmonizing aesthetic excellence with a methodical development

approach, we aspire to deliver a template that not only meets but exceeds the expectations of user.

FILE STRUCTURE :

```
weather_app/ |  
|   current.html  
|   forecast.html  
|   index.html  
|   script.js  
└   style.css
```

1.3 HTML Code :

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>weather app</title>  
    <meta name="viewport" content="width=device-width, initial-  
scale=1">  
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/  
css/bootstrap.min.css" rel="stylesheet" integrity="sha384-  
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH  
" crossorigin="anonymous">  
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/  
font-awesome/4.4.0/css/font-awesome.min.css">  
    <link rel="stylesheet" type="text/css" href="style.css">  
</head>  
  
<body>  
    <nav class="navbar navbar-expand-lg bg-body-tertiary"  
id="nav_bar">  
        <div class="container-fluid mx-auto" width="70vw">  
            <a class="navbar-brand fs-1"  
href="index.html"><strong>Weather App</strong></a>
```

```
<button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarScroll" aria-
controls="navbarScroll" aria-expanded="false" aria-label="Toggle
navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarScroll">
    <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-
scroll" style="--bs-scroll-height: 100px;">
        <li class="nav-item">
            <a class="nav-link fs-3" aria-current="page"
href="index.html">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link fs-3"
href="current.html">Weather</a>
        </li>
        <li class="nav-item">
            <a class="nav-link fs-3"
href="forecast.html">Forecast</a>
        </li>
    </ul>
</div>
</div>
</nav>

<div class="vw-100 p-5" style="background-image: url('/img/
1056-1200x1200.jpg'); background-size: cover; min-height: 82vh;
height: fit-content;">
    <h2 class="text-center">Get Current Weather Information</h2>
    <div class="container-sm">
        <div class="col-md-12">
            <div class="row form-group form-inline" id="city">
                <div id="error"></div>
                <input type="text" name="cityname" id="cityname"
class="form-control text-center" placeholder="City name"></
input>
                <button class="btn btn-primary" id="submitCity"
onclick="getWeather(document.getElementById('cityname').value);"
>Search</button>
                <div id="weather-data" class="mx-auto">
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```

</div>
<ul class="nav justify-content-center my-4">
    <li class="nav-item fs-4 mx-2">Developed by Uday Kalra</li>
</ul>

<script
    src="https://code.jquery.com/jquery-3.2.1.min.js"
    integrity="sha256-
hwg4gsxgFZhOsEEamd0YGBf13FyQuiTwLAQgxVSNg4="
    crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvygol6VFvRkX/vR+Vc4jQkC+hVqc2pM80Dewa9r"
    crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
    integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCs0Gab
    YfZo0T0to5eqruptLy" crossorigin="anonymous"></script>

<script type="text/javascript" src="script.js"></script>
</body>
</html>

```

```

<!DOCTYPE html>
<html>
<head>
    <title>Weather app</title>
    <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/
css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH
" crossorigin="anonymous">
        <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-body-tertiary"
id="nav_bar">
        <div class="container-fluid mx-auto" width="70vw">
            <a class="navbar-brand fs-1"
href="index.html"><strong>Weather App</strong></a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarScroll" aria-

```

```
controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarScroll">
    <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scroll-height: 100px;">
        <li class="nav-item">
            <a class="nav-link fs-3" aria-current="page" href="index.html">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link fs-3" href="current.html">Weather</a>
        </li>
        <li class="nav-item">
            <a class="nav-link fs-3" href="forecast.html">Forecast</a>
        </li>
    </ul>
</div>
</div>
</nav>

<div class="vw-100 p-5" style="background-image: url('/img/1056-1200x1200.jpg'); background-size: cover; min-height: 82vh; height: fit-content;">
    <h2 class="text-center">Get Weather Information upto 16 days</h2>
    <div class="container-sm">
        <div class="col-md-12">
            <div class="row form-group form-inline" id="city">
                <div id="error"></div>
                <input type="text" name="cityname" id="cityname" class="form-control text-center" placeholder="City name"></input>
                <input type="text" name="days" id="days" class="form-control text-center" placeholder="Days"></input>
                <button class="btn btn-primary" id="submitCity" onclick="getForecast(document.getElementById('cityname').value, document.getElementById('days').value);">Get Forecast</button>
            </div>
        </div>
    </div>
</div>
<div class="table-responsive">
    <table class="table table-bordered table-condensed">
```

```

<thead>
  <tr>
    <th>Icon</td>
    <th>Weather</td>
    <th>Description</td>
    <th>Morning</td>
    <th>Night</td>
    <th>Min</td>
    <th>Max</td>
    <th>Pressure</td>
    <th>Humidity</td>
    <th>Wind Speed</td>
  </tr>
</thead>
<tbody id="forecast">

  </tbody>
</table>
</div>
</div>
<ul class="nav justify-content-center my-4">
  <li class="nav-item fs-4 mx-2">Developed by Uday Kalra</li>
</ul>

<script src="https://code.jquery.com/jquery-3.2.1.min.js"
integrity="sha256-hwg4gsxgFZhOsEEamd0YGBf13FyQuiTwlAQgxVSNgt4="
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/
core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/
ismYTF4hNIPjVp/Zjvyol6VFvRkX/vR+Vc4jQkC+hVqc2pM80Dewa9r"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/
dist/js/bootstrap.min.js"
integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCs0Gab
YfZo0T0to5eqruptLy" crossorigin="anonymous"></script>
<script type="text/javascript" src="script.js"></script>
</body>
</html>

<!DOCTYPE html>
<html>
<head>
  <title>weather app</title>
  <meta name="viewport" content="width=device-width, initial-
scale=1">

```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-body-tertiary" id="nav_bar">
        <div class="container-fluid mx-auto" width="70vw">
            <a class="navbar-brand fs-1" href="index.html"><strong>Weather App</strong></a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarScroll" aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarScroll">
                <ul class="navbar-nav me-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scroll-height: 100px;">
                    <li class="nav-item">
                        <a class="nav-link fs-3" aria-current="page" href="index.html">Home</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link fs-3" href="current.html">Weather</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link fs-3" href="forecast.html">Forecast</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="vw-100 d-flex justify-content-center align-items-center" style="background-image: url('/img/1056-1200x1200.jpg'); background-size: cover; min-height: 82vh; height: fit-content;">
        <div class="container-mid p-5 mx-auto mh-50" id="colText">
            <h2 class="text-center" id="tagline">WEATHER</h2>
            <h2 class="text-center" id="tagline-2">Get weather information of any city.</h2>
        </div>
    </div>
</body>
```

```

<ul class="nav justify-content-between my-4">
    <li class="nav-item fs-5 mx-2">Temperature</li>
    <li class="nav-item fs-5 mx-2">Pressure</li>
    <li class="nav-item fs-5 mx-2">Wind Speed</li>
    <li class="nav-item fs-5 mx-2">Humidity</li>
</ul>

    <button type="button" class="btn btn-primary"><a href="current.html" style="text-decoration: none; color: #fff;">CHECK WEATHER</a></button>
    <button type="button" class="btn btn-primary"><a href="forecast.html" style="text-decoration: none; color: #fff;">GET FORECAST</a></button>
</div>
</div>
<ul class="nav justify-content-center my-4">
    <li class="nav-item fs-4 mx-2">Developed by Uday Kalra</li>
</ul>

<script src="https://code.jquery.com/jquery-3.2.1.min.js" integrity="sha256-hwg4gsxgFZhOsEEamd0YGBf13FyQuiTwlAQgxVSNgt4=" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js" integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvygol6VFvRkX/vR+Vc4jQkC+hVqc2pM80Dewa9r" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js" integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCs0GabYfZo0T0to5eqruptLy" crossorigin="anonymous"></script>
</body>
</html>

```

CSS CODE :

```

@import url('https://fonts.googleapis.com/css?family=Indie+Flower|Raleway');

body{
    font-family: 'Raleway', serif;
    background-color: white;

```

```
        margin: 0;
        padding: 0;
    }
.container-fluid{
    width: fit-content;
}
#tagline{
    color: black;
    padding-top: 4%;
    font-size: 2.3em;
}
#tagline-2{
    color: black;
    padding-top: 2%;
    padding-bottom: 5%;
    font-size: 2em;
}
#city{
    width: 60%;
    margin: auto;
    text-align: center;
}
input[type='text'],#submitCity{
    height: 1.8em;
    font-size: 1.6em;
    margin-bottom: 8%;
}
table{
    font-size: 1.6em;
}
#weather-data, .container-mid{
    min-width:200px;
    max-width: 40vw;
    background-color: #fff;
    text-align: center;
    border-radius: 5px;
}
#weather-data h3{
    padding-top: 3%;
}
#weather-data p:last-child{
    padding-bottom: 3%;
}
```

JAVASCRIPT CODE :

```
function getWeather(city){
    if (city) {
        response = fetch("http://api.openweathermap.org/data/
2.5/weather?q=" + city +
"&units=metric&appid=932bb81b0aa37e297a52991b19f5f815")
            .then(response => response.json())
            .then(data => {
                var formattedData=formatWeather(data);
                document.getElementById("weather-
data").innerHTML=formattedData;
                document.getElementById('cityname').value="";
            })
            .catch(error => {
                var error='\
<div class="alert alert-danger" role = "alert" >
\
                    City not found! \
                    <button type="button" class="btn-close"
data-bs-dismiss="alert" aria-label="Close"></button> \
                </div>'
                document.getElementById('error').innerHTML=error;
            });
    } else{
        var error = '\
<div class="alert alert-danger" role = "alert" > \
            You must enter a city name! \
            <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button> \
        </div>'
        document.getElementById('error').innerHTML=error;
    }
}

function formatWeather(data){
    return "<h3>Current Weather for " + data.name + ", " +
data.sys.country + "</h3>" +
    "<p>Weather: " + data.weather[0].main+ "</p>" +
    "<p>Weather Description: " +
data.weather[0].description +"<img src='http://
```

```

openweathermap.org/img/w/" + data.weather[0].icon + ".png' />" +
"</p>" +
    "<p>Temperature: " + data.main.temp + "&deg;C</p>" +
    "<p>Pressure: " + data.main.pressure + "hPa</p>" +
    "<p>Humidity: " + data.main.humidity + "%</p>" +
    "<p>Min Temperature: " + data.main.temp_min +
"&deg;C</p>" +
    "<p>Max Temperature: " + data.main.temp_max +
"&deg;C</p>" +
    "<p>Wind Speed: " + data.wind.speed + "m/s</p>";
}

function getForecast(city, days) {
    if (city) {
        response = fetch("http://api.openweathermap.org/data/
2.5/forecast/daily?q=" + city + "&cnt=" + days +
"&units=metric&appid=d610395e85b50074b834a0234b0776db")
            .then(response => response.json())
            .then(data => {
                var formattedData=formatForecast(data);

document.getElementById("forecast").innerHTML=formattedData;
                document.getElementById('cityname').value="";
                document.getElementById('days').value=""
            })
            .catch(error => {
                var error='\
                    <div class="alert alert-danger" role = "alert" > \
                        City not found! \
                        <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button> \
                    </div>'
                document.getElementById('error').innerHTML=error;
            });
    } else {
        var error = '\
            <div class="alert alert-danger" role = "alert" > \
                You must enter a valid city name and Number of day \
should be greater than 0 and less than or equal to 16. \
                <button type="button" class="btn-close" data-bs-
dismiss="alert" aria-label="Close"></button> \
            </div>'
        document.getElementById('error').innerHTML=error;
    }
}

```

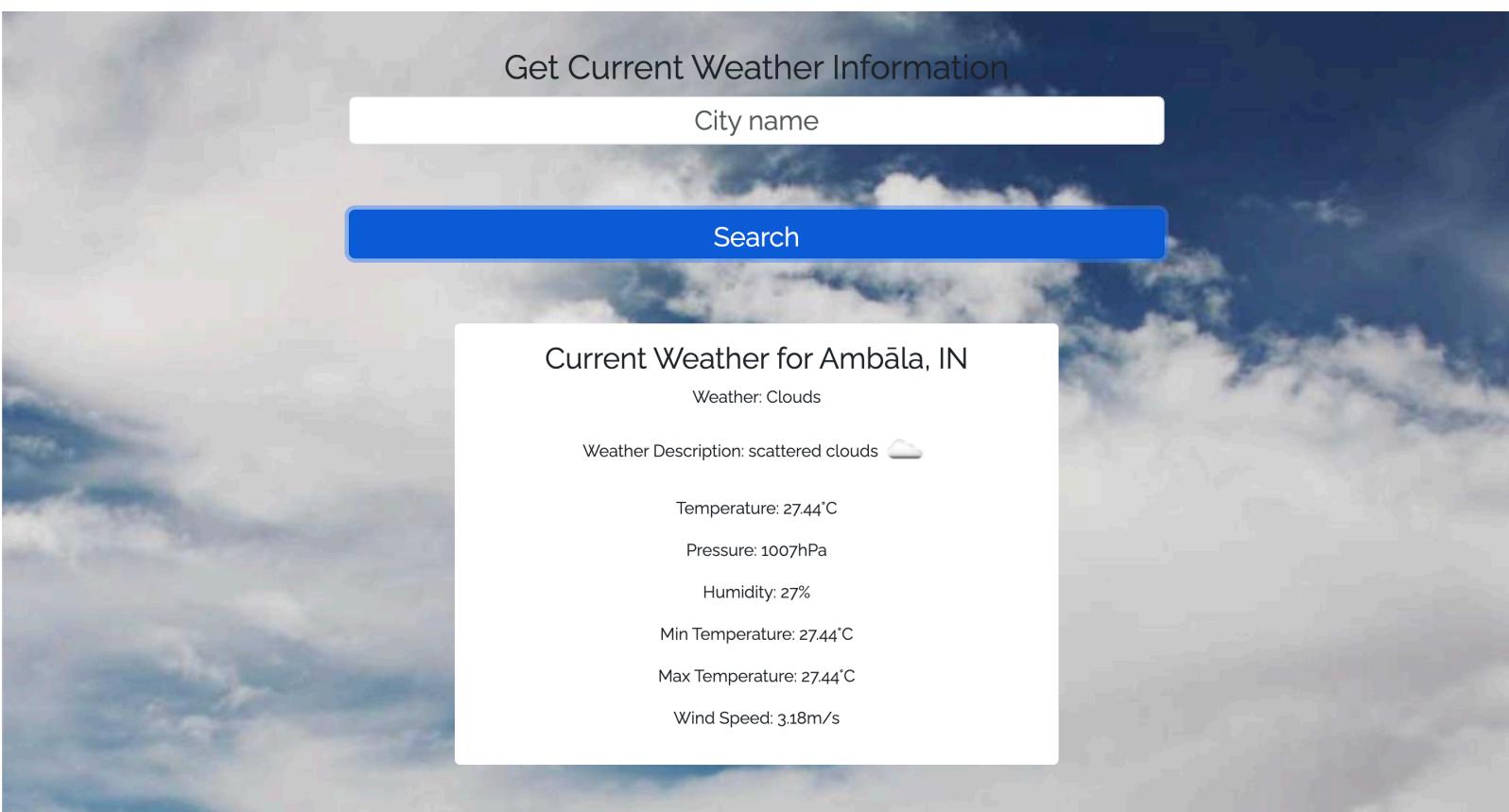
```
}

function formatForecast(data){
    var table="";
    for (var i = 0; i < data.list.length; i++) {
        table += "<tr>";
        table += "<td><img src='http://openweathermap.org/img/w/" + data.list[i].weather[0].icon + ".png' /></td>";
        table += "<td>" + data.list[i].weather[0].main + "</td>";
        table += "<td>" + data.list[i].weather[0].description + "</td>";
        table += "<td>" + data.list[i].temp.morn + "&deg;C</td>";
        table += "<td>" + data.list[i].temp.night + "&deg;C</td>";
        table += "<td>" + data.list[i].temp.min + "&deg;C</td>";
        table += "<td>" + data.list[i].temp.max + "&deg;C</td>";
        table += "<td>" + data.list[i].pressure + "hPa</td>";
        table += "<td>" + data.list[i].humidity + "%</td>";
        table += "<td>" + data.list[i].speed + "m/s</td>";
        table += "</tr>";
    }
    return table;
}
```

Results:

Snippets of the project attached below:

4.1 Desktop view:



Weather App

[Home](#) [Weather Forecast](#)

Get Current Weather Information

Search

Developed by Uday Kalra

Weather App

[Home](#) [Weather Forecast](#)

Get Weather Information upto 16 days

Get Forecast

Icon	Weather	Description	Morning	Night	Min	Max	Pressure	Humidity	Wind Speed
------	---------	-------------	---------	-------	-----	-----	----------	----------	------------

Developed by Uday Kalra