

PROFESSIONAL TRAINING REPORT

Sathyabama Institute of Science and Technology (Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of Bachelor
of Technology in Information Technology

By

JOSHVA J
REG. NO. 39120048



DEPARTMENT OF INFORMATION TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

NOVEMBER 2021

	SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY) Accredited with Grade “A” by NAAC (Established under Section 3 of UGC Act, 1956) JEPPIAAR NAGAR, RAJIV GANDHI SALAI CHENNAI- 600119 www.sathyabama.ac.in	
---	---	---

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **JOSHVA J (Reg. No: 39120048)** who carried out the project entitled “**WEATHER-APPLICATION using React-JS**” under my supervision from June 2021 to November 2021.

Internal Guide

Dr . P. Jeyanthi

Head of the Department

Dr. R. Subashini,M.E.,Ph.D.

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, **JOSHVA J** hereby declare that the project report entitled **WEATHER-APPLICATION using React-JS** done by me under the guidance of **Dr . P. Jeyanthi** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D. Dean**, School of Computing, **Dr. Dr. R. Subhashini, M.E., Ph.D. Heads of the Department of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. P. JEYANTHI, M.E., Ph.D.** for his valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

Weather is the state of the atmosphere at a given place and time in regards to heat, cloudiness, dryness, sunshine, wind, and rain. Of all the geophysical phenomena weather is the most significant one that influences us. Weather can vary greatly and largely depends on climate, seasons and various other factors. The chief goal of this work is to get the weather forecast of any city throughout the world through an application. This paper aims at creating a web application using javascript framework React-JS.

This includes temperature, location, precipitation, wind speed, and humidity. This report provides a basic understanding of the purpose and scope of weather, the basic principles and the general models developed for weather.

	TABLE OF CONTENTS	
CHAPTER No.	TITLE	PAGE No
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	LIST OF ABBREVIATIONS	iv
1	INTRODUCTION	1
	1.1 Objective of the project	1
	1.1.1 Necessity	1
	1.1.2 Software development method	1
	1.1.3 Layout of the document	
	1.2 Web Design Overview	2
2.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	2
	2.1 Project Proposal	2
	2.1.1 Mission	2
	2.1.2 Goal	3
	2.2 Scope of the project	3
	2.3 Overview of the project	3
	2.4 Flow Chart	3

3. EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED 4

	3.1 System Study	4
	3.1.1 System requirements specifications	4
	3.2 System Specifications	4
	3.2.1 JavaScript Overview	4
	3.2.2 React Overview	5

	3.3 Steps to download & install React-js	5
	3.3.1 What is create-react-app	6
	3.4 Packages to be installed	7
	3.5 Modules	7
	3.6 Module Details	7
4.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	8
	4.1 Performance Analysis	8
	4.2 Discussion	9
5	SUMMARY AND CONCLUSIONS	9
	5.1 Summary	9
	5.2 Conclusion	9
	REFERENCES	9
	APPENDIX	10
	A. SCREENSHOTS	10
	B. SOURCE CODE	12

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
2.4	Flow Chart Of the Project	3
5.1	Weather app	10
5.2	weather info	10

CHAPTER-1

INTRODUCTION

1.1 OBJECTIVE OF THE PROJECT:

This project objective is to deliver how the present state of the atmosphere will be displayed.

1.1.1 *Necessity:*

This website helps to know the weather condition at different areas. This Application is very easy to use. It can work accurately and very smoothly. In aspects of time value, it is worthy. In this website the user should enter the area location to know the condition and depending on the weather the citizens can travel and do their work. And also the user can check the weather of different zones and help others to know how weather is changing. Therefore, user will get the idea of weather and he/she can know the humidity and wind-speed easily.

1.1.2 *Layout of the document:*

This documentation starts with formal introduction. After introduction analysis and design of the project are described. In analysis and design of the project have many parts such as project proposal, mission, goal, target audience, environment. After that design and table diagram will be found. Use cases and test cases are in chapter 2 and chapter 3 respectively. Finally, this documentation finished with result and Conclusion part.

1.2 Web Design Overview:

The React design website process starts with visual concept which includes a HOC component that wraps another component by adding extra functionalities or extra properties to the design pattern like stateless functions,

render props, controlled components, condition rendering and react hooks for the design of the website. However, this is our abstract for some commonly used logics and by adding extra patterns to the code for the desired website.

CHAPTER-2

AIM AND SCOPE OF THE PRESENT INVESTIGATION

2.1 Project Proposal:

The project proposal is the term of documents. A project can describe the project proposal. It is the set of all plans of a project. Like, how the software works, what are the steps to complete the entire projects, and what are the software requirements and analysis for this project. In my project, I am doing all the steps and also risk and reward and other project dependencies in the project proposal.

2.1.1 Mission:

An online web-based application is very popular and well known to everyone. Now a day's everybody wants to get it and work with it. This application runs everywhere in the world. It was originally built for everyone. This project brings every small application into one website which will help everyone for their works easily. The weather application helps to know the condition at different parts of the world. Just by entering the name of the city and it display's information like temperature, humidity, wind-speed, and the location of the place present in. This web app helps the user to know the weather across the globe.

2.1.2 Goal:

The goal of my project is to build a web-based application for understanding the climate at specific location.

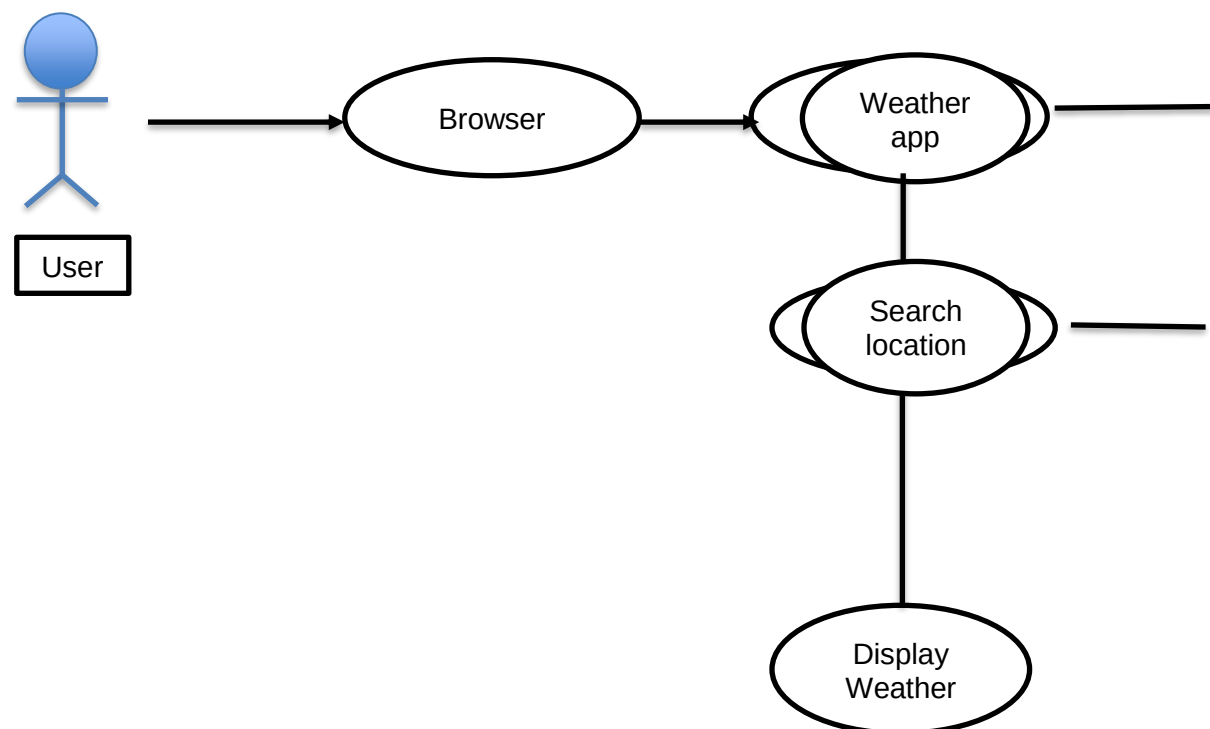
2.2 Scope of the Project:

I want to build a web-based application for everyone in the society which will be very easy to know about the climate. It works with a common server or browser which is very familiar with us. They cannot change the customizations of fonts and colors of the web application. I want to focus on the application itself and the features of that application that help build a good management community.

2.3 Overview of the Project:

The overview of the project is to provide a web-based application to the user. Therefore, the user can directly access the weather application easily from any place and in any device just by searching places.

2.4 Flow Chart:



CHAPTER-3

EXPERIMENTAL OR MATERIAL AND METHODS

3.1 System Study:

To develop this model we use new modern technology like React, JavaScript for overcoming our problem statement.

3.1.1 System requirement specifications:

a) Hardware requirements:

- Processor : Intel Pentium D 2.8 GHZ.
- RAM : 6GB
- Hard Disk : 80GB

b) Software requirements:

- OS : Windows.
- Framework : React, JavaScript.
- Web Browser : Chrome.
- Code editor : Visual Studio Code.

3.2 System Specifications:

3.2.1 JavaScript Overview:

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. JavaScript is a lightweight, interpreted programming language. JavaScript is the most popular programming language in the world and that makes it a programmer's great choice. Once you learnt Javascript, it helps you developing great front-end as well as back-end software's using different Javascript based frameworks like React, Node.JS etc. Javascript helps in manipulating HTML page on the fly. This helps in adding and deleting any HTML tag very easily using javascript and modify your HTML to change its look and feel based on different devices and requirements.

3.2.2 React Overview:

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It's 'V' in MVC. ReactJS is an open-source, component-based front-end library responsible only for the view layer of the application. It is maintained by Facebook.

React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application, and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug.

A React application is made of multiple components, each responsible for rendering a small, reusable piece of HTML. Components can be nested within other components to allow complex applications to be built out of simple building blocks. A component may also maintain an internal state – for example, a Tab List component may store a variable corresponding to the currently open tab.

3.3 Steps to Download & Install ReactJS:

- ❖ First of all, you are going to need NPM (or Yarn, alternatively). Let's use NPM for this example.
- ❖ If you don't have it installed on your system, then you need to head to the official Node.js website to download and install Node, which also includes NPM (Node Package Manager).
- ❖ Select the "Recommended For Most Users" button and download the current version for your operating system.
- ❖ After you download and install Node, start your terminal/command prompt and run `node -v` and `npm -v` to see which versions you have.
- ❖ Your version of NPM should be at least 5.2.0 or newer because create-react-app requires that we have NPX installed.
- ❖ If you have an older version of NPM, run this command to update it:

npm install -g npm

3.3.1 What is create-react-app?

Since it is complicated and takes a lot of time, we don't want to configure React manually. create-react-app is a much easier way which does all the configuration and necessary package installations for us automatically and starts a new React app locally, ready for development.

Another advantage of using create-react-app is that you don't have to deal with Babel or Webpack configurations. All of the necessary configurations will be made by create-react-app for you.

How to Install Create-React-App

In order to install your app, first go to your workspace (desktop or a folder) and run the following command:

npx create-react-app my-app

The installation process may take a few minutes. After it is done, you should see a folder that appears in your workspace with the name you gave to your app.

Note: If you're on Mac and receiving permission errors, don't forget to be a super user first with the sudo command.

How to Run the App You Created with Create-React-App

After the installation is completed, change to the directory where your app was installed:

cd my-app

and finally run **npm start** to see your app live on localhost:

npm start

If you see something REACT logo in your browser, you are ready to work with React.

3.4 Packages to be installed:

- ☒ npm install react
- ☒ npm install Axios

3.5 Weather stack:

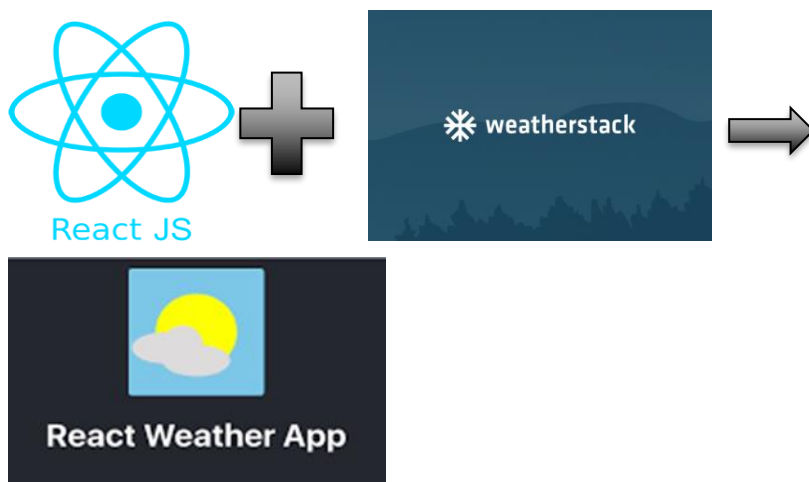
The weather stack API was built to deliver accurate weather data for any application and use case. Our straightforward API design will make it easy to use the API.

The first step to using the API is to authenticate with your weather stack account's unique API access key, To authenticate with the API, simply use the base URL below and pass your API access key to the API's access key parameter.

3.6 Flex Grid :-

A grid system based on flex display property.

The grid should be displayed on the app to divided the app in to two parts Nav bar and the display bar which contains the information present in data under the console. The data will be printed accordingly.



CHAPTER-4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

4.1 Performance Analysis:

Website performance optimization, the focal point of technologically superior website designs is the primary factor dictating success for modern daily needs. After all, unimpressive website performance kills admission process when the torture of waiting for slow Web pages to load frustrates visitors into seeking alternatives – impatience is a digital virtue!

We created the following six chapter in-depth speed optimization guide to show you how important it is to have a fast loading, snappy website! Countless research papers and benchmarks prove that optimizing your sites' speed is one of the most affordable and highest ROI providing investments!

Lightning-fast page load speed amplifies visitor engagement, retention, and boosts sales. Instantaneous website response leads to higher conversion rates, and every 1 second delay in page load decreases customer satisfaction by 16 percent, page views by 11 percent and conversion rates by 7 percent according to a recent Aberdeen Group research.

4.2 Discussion:

While discussions provide avenues for exploration and discovery, leading a discussion can be anxiety-producing: discussions are, by their nature, unpredictable, and require us as instructors to surrender a certain degree of control over the flow of information. Fortunately, careful planning can help us ensure that discussions are lively without being chaotic and exploratory without losing focus. When planning a discussion, it is helpful to consider not only cognitive, but also social/emotional, and physical factors that can either foster or inhibit the productive exchange of ideas.

CHAPTER-5

SUMMARY AND CONCLUSIONS

5.1 Summary:

This project objective is to deliver the online website that brings every useful application into one platform. This project is an attempt to provide the advantage of all small application in to one platform that help the user in time management. It helps the user in critical situations.

5.2 Conclusion:

This website is helpful for people to identify tasks that are written in todo-list(notepad). So, no one will forget any important messages by seeing them and also it helps in calculating mathematical operations using the calculator and by using the BMI calculator helps in Gym for calculating your BMI directly. So, this Website is useful for time management and it is very easy to use.

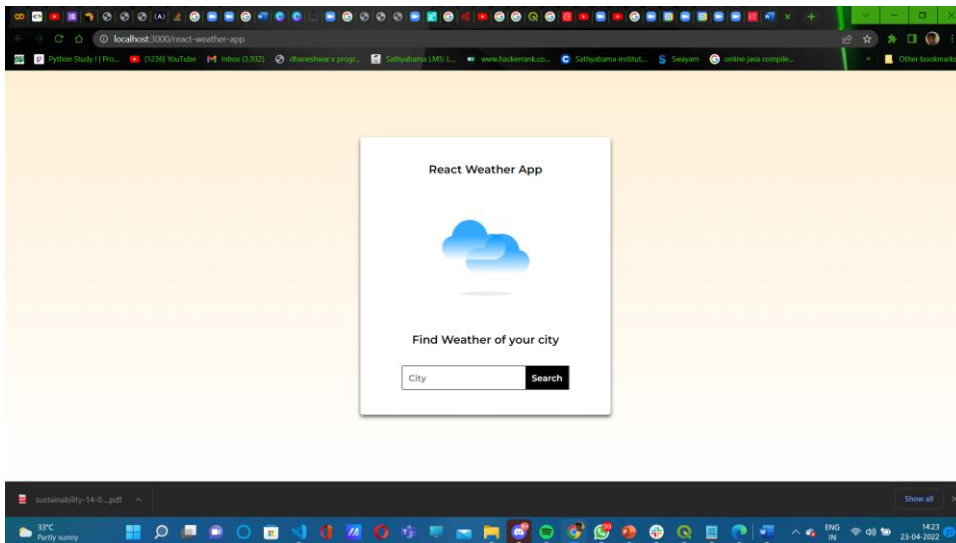
REFERENCES:

- ❖ <https://github.com>
- ❖ <https://stackoverflow.com>
- ❖ <https://www.geeksforgeeks.org>
- ❖ <https://www.algofusion.org>

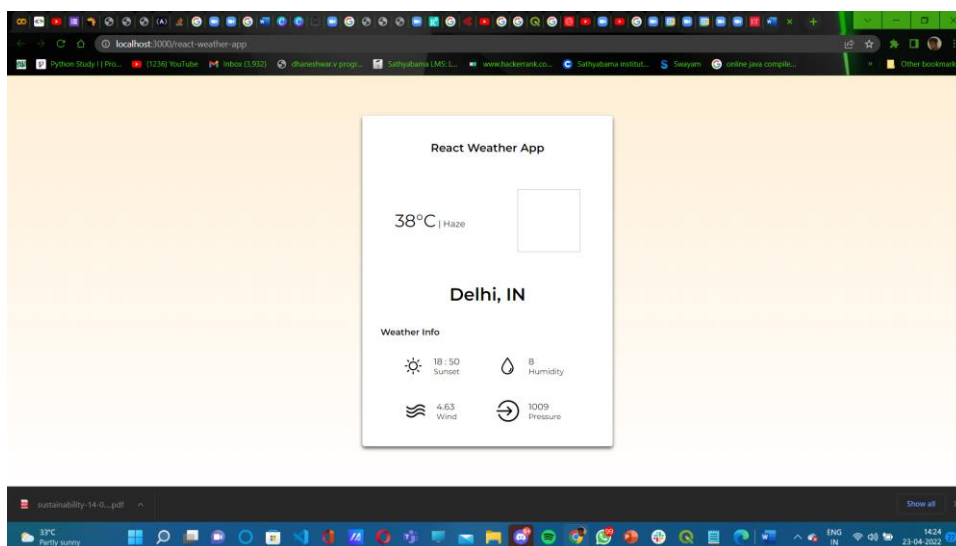
APPENDIX:

A. SCREENSHOTS

5.1 weather:



5.2 Weather info:



B. SOURCE CODE:

Public:

public\index.html:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link
href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;500;600&dis
play=swap" rel="stylesheet">
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the
build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
```

```

<div id="root"></div>
<!--
  This HTML file is a template.
  If you open it directly in the browser, you will see an empty page.

  You can add webfonts, meta tags, or analytics to this file.
  The build step will place the bundled scripts into the <body> tag.

  To begin the development, run `npm start` or `yarn start`.
  To create a production bundle, use `npm run build` or `yarn build`.
-->
</body>
</html>

```

src:

Src\index.js

```

import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';

ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

```

src\index.css:

```
body {
  margin: 0;
  background: linear-gradient(papayawhip, white);
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
  font-family: source-code-pro, Menlo, Monaco, Consolas, 'Courier New',
  monospace;
}
```

src\App.js:

```
import React, { useState } from "react";
import styled from "styled-components";
import Axios from "axios";
import CityComponent from "./modules/CityComponent";
import WeatherComponent from "./modules/WeatherInfoComponent";

export const WeatherIcons = {
  "01d": "/react-weather-app/icons/sunny.svg",
  "01n": "/react-weather-app/icons/night.svg",
  "02d": "/react-weather-app/icons/day.svg",
  "02n": "/react-weather-app/icons/cloudy-night.svg",
  "03d": "/react-weather-app/icons/cloudy.svg",
  "03n": "/react-weather-app/icons/cloudy.svg",
  "04d": "/react-weather-app/icons/perfect-day.svg",
  "04n": "/react-weather-app/icons/cloudy-night.svg",
  "09d": "/react-weather-app/icons/rain.svg",
```

```

    "09n": "/react-weather-app/icons/rain-night.svg",
    "10d": "/react-weather-app/icons/rain.svg",
    "10n": "/react-weather-app/icons/rain-night.svg",
    "11d": "/react-weather-app/icons/storm.svg",
    "11n": "/react-weather-app/icons/storm.svg",
  };

```

```

const Container = styled.div`
  display: flex;
  flex-direction: column;
  align-items: center;
  width: 380px;
  padding: 20px 10px;
  margin: auto;
  border-radius: 4px;
  box-shadow: 0 3px 6px 0 #555;
  background: white;
  font-family: Montserrat;
`;

```

```

const AppLabel = styled.span`
  color: black;
  margin: 20px auto;
  font-size: 18px;
  font-weight: bold;
`;

```

```

const CloseButton = styled.span`
  padding: 2px 3px;
  background-color: black;
  border-radius: 50%;
  color: white;
  position: absolute;
`;

```

```

function App() {
  const [city, updateCity] = useState();
  const [weather, updateWeather] = useState();
  const fetchWeather = async (e) => {
    e.preventDefault();
    const response = await Axios.get(
      `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=fe4feefa8543e06d4f3c66d92c61b69c`,

```

```

    );
    updateWeather(response.data);
  };
  return (
    <Container>
      <AppLabel>React Weather App</AppLabel>
      {city && weather ? (
        <WeatherComponent weather={weather} city={city} />
      ) : (
        <CityComponent updateCity={updateCity} fetchWeather={fetchWeather} />
      )}
    </Container>
  );
}

export default App;

```

src\components\citycomponenta.js:

```

import styled from "styled-components";
import React from "react";

```

```

const SearchBox = styled.form`
  display: flex;
  flex-direction: row;
  justify-content: space-evenly;
  margin: 20px;
  border: black solid 1px;
  border-radius: 2px;

```

```

  & input {
    padding: 10px;
    font-size: 14px;
    border: none;
    outline: none;
    font-family: Montserrat;
    font-weight: bold;
  }
  & button {
    background-color: black;
    font-size: 14px;
    padding: 0 10px;

```

```

        color: white;
        border: none;
        outline: none;
        cursor: pointer;
        font-family: Montserrat;
        font-weight: bold;
    }
`;
const ChooseCityLabel = styled.span`
    color: black;
    margin: 10px auto;
    font-size: 18px;
    font-weight: bold;
`;
const WelcomeWeatherLogo = styled.img`
    width: 140px;
    height: 140px;
    margin: 40px auto;
`;
const CityComponent = (props) => {
    const { updateCity, fetchWeather } = props;
    return (
        <>
            <WelcomeWeatherLogo src={"/react-weather-app/icons/perfect-day.svg"} />
            <ChooseCityLabel>Find Weather of your city</ChooseCityLabel>
            <SearchBox onSubmit={fetchWeather}>
                <input
                    onChange={(e) => updateCity(e.target.value)}
                    placeholder="City"
                />
                <button type="submit">Search</button>
            </SearchBox>
        </>
    );
};
export default CityComponent;

```

src\components\weatherinfo\components.js:

```

import React from "react";
import styled from "styled-components";
import { WeatherIcons } from "../App";

```

```

export const WeatherInfoIcons = {
  sunset: "/react-weather-app/icons/temp.svg",
  sunrise: "/react-weather-app/icons/temp.svg",
  humidity: "/react-weather-app/icons/humidity.svg",
  wind: "/react-weather-app/icons/wind.svg",
  pressure: "/react-weather-app/icons/pressure.svg",
};

const Location = styled.span`
  margin: 15px auto;
  text-transform: capitalize;
  font-size: 28px;
  font-weight: bold;
`;

const Condition = styled.span`
  margin: 20px auto;
  text-transform: capitalize;
  font-size: 14px;
  & span {
    font-size: 28px;
  }
`;

const WeatherInfoLabel = styled.span`
  margin: 20px 25px 10px;
  text-transform: capitalize;
  text-align: start;
  width: 90%;
  font-weight: bold;
  font-size: 14px;
`;

const WeatherIcon = styled.img`
  width: 100px;
  height: 100px;
  margin: 5px auto;
`;

const WeatherContainer = styled.div`
  display: flex;
  width: 100%;
  margin: 30px auto;
  flex-direction: row;
  justify-content: space-between;
  align-items: center;
`;

const WeatherInfoContainer = styled.div`
  display: flex;

```

```

    width: 90%;
    flex-direction: row;
    justify-content: space-evenly;
    align-items: center;
    flex-wrap: wrap;
  `;
const InfoContainer = styled.div`
  display: flex;
  margin: 5px 10px;
  flex-direction: row;
  justify-content: space-evenly;
  align-items: center;
`;
const InfoIcon = styled.img`
  width: 36px;
  height: 36px;
`;
const InfoLabel = styled.span`
  display: flex;
  flex-direction: column;
  font-size: 14px;
  margin: 15px;
  & span {
    font-size: 12px;
    text-transform: capitalize;
  }
`;

const WeatherInfoComponent = (props) => {
  const {name, value} = props;
  return (
    <InfoContainer>
      <InfoIcon src={WeatherInfoIcons[name]}/>
      <InfoLabel>
        {value}
        <span>{name}</span>
      </InfoLabel>
    </InfoContainer>
  );
};

const WeatherComponent = (props) => {
  const {weather} = props;
  const isDay = weather?.weather[0].icon?.includes('d')
  const getTime = (timeStamp) => {
    return `${new Date(timeStamp * 1000).getHours()} : ${new
Date(timeStamp * 1000).getMinutes()}`
  }

```

```

    }
    return (
      <>
        <WeatherContainer>
          <Condition>
            <span>`${Math.floor(weather?.main?.temp -
273)}°C`</span>
            {` | ${weather?.weather[0].description}`}
          </Condition>
          <WeatherIcon src={WeatherIcons[weather?.weather[0].icon]}/>
        </WeatherContainer>
        <Location>`${weather?.name},
${weather?.sys?.country}`</Location>

        <WeatherInfoLabel>Weather Info</WeatherInfoLabel>
        <WeatherInfoContainer>
          <WeatherInfoComponent name={isDay ? "sunset" : "sunrise"}
                                value={`${getTime(weather?.sys[isDay ?
"sunset" : "sunrise"])} `}/>
          <WeatherInfoComponent name={"humidity"}
value={weather?.main?.humidity}/>
          <WeatherInfoComponent name={"wind"}
value={weather?.wind?.speed}/>
          <WeatherInfoComponent name={"pressure"}
value={weather?.main?.pressure}/>
        </WeatherInfoContainer>
      </>
    );
  };
};

export default WeatherComponent;

```

