**Project Report**

**Title:** Weather Forecasting application using MERN stack.

**1. Introduction:**

The Weather Forecasting Application is a web-based tool designed to provide users with real-time weather information and forecasts for specific locations. This project integrates with the OpenWeatherMap API to retrieve weather data and displays it in a user-friendly interface. The application allows users to search for weather forecasts by city name and view both current weather conditions and future forecasts.

**2. Objectives:**

* Develop a web application for weather forecasting.
* Integrate with the OpenWeatherMap API to fetch weather data.
* Display current weather information and forecasts for specified locations.
* Provide a user-friendly interface for easy navigation and interaction.

**3. Methodology:**

* **Backend:** Implemented using Node.js and Express.js framework to create a RESTful API.
* **Frontend:** Built using React.js library to create a dynamic user interface.
* **API Integration:** Utilized Axios library to make HTTP requests to the OpenWeatherMap API.
* **Styling:** Applied CSS for styling components and enhancing the visual appeal of the application.

**4. Code:**

**Backend**

import express from 'express';

import axios from 'axios';

import cors from 'cors';

import \_ from 'lodash';

const app = express();

const port = 3001;

const hostname = '127.0.0.1';

const apiKey = "16c4c78266a7343d3cd9e75b95ee98e1";

let country\_code = "IN";

let city\_name = "Delhi";

app.use(express.json());

app.use(cors());

app.get('/api/weather', async (req, res) => {

try {

let place = await axios.get(`http://api.openweathermap.org/geo/1.0/direct?q=${city\_name},${country\_code}&limit=1&appid=${apiKey}`);

if (\_.isEmpty(place.data)) {

console.log("No match found");

city\_name = "Delhi";

return res.json({

error: "No match found"

})

}

const lat = place.data[0].lat;

const lon = place.data[0].lon;

let currentWeather = axios.get(`https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&appid=${apiKey}&units=metric`);

let forecast = axios.get(`https://api.openweathermap.org/data/2.5/forecast?lat=${lat}&lon=${lon}&appid=${apiKey}&units=metric`);

const result = await Promise.all([currentWeather, forecast]);

// result[0] => current weather data

// result[1] => forecast data

return res.json({

weather: result[0].data,

place: place.data[0],

forecasts: result[1].data.list

});

} catch (err) {

console.log(err);

res.status(500).json({ error: err.message });

}

});

app.post("/api/search", async (req, res) => {

try {

city\_name = req.body.search;

res.redirect("/api/weather");

} catch (err) {

console.log(err);

res.status(500).json({ error: err.message });

}

});

app.listen(port, () => {

console.log('Listening at http://' + hostname + ":" + port);

});

**Frontend**

//main.jsx

import React from 'react'

import ReactDOM from 'react-dom/client'

import App from './App.jsx'

import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(

<React.StrictMode>

<App />

</React.StrictMode>,

)

//App.jsx

import { useEffect, useState } from 'react'

import bgImg from './assets/bg4.jpg'

import \_ from 'lodash'

import axios from 'axios'

import './App.css'

import Time from './time'

import DateComponent from './Date'

function App() {

const [weatherData, setWeatherData] = useState({});

const [error, setError] = useState(null);

useEffect(() => {

axios.get('http://127.0.0.1:3001/api/weather')

.then((result) => {

setWeatherData(result.data);

})

.catch((error) => {

console.error(error);

setError(error);

})

}, []);

return (

<>

<div className="bg">

<img src={bgImg} alt="" />

</div>

<div className="container">

<div className="box">

<SearchBar />

<div className="head">

<h1>Today's Weather</h1>

<DateComponent />

</div>

<div className="main">

{weatherData && weatherData.place && (

<>

<p className="info">

{weatherData.place.name}

</p>

<span>

Min: {weatherData.weather.main.temp\_min}&deg;C | Max: {weatherData.weather.main.temp\_max}&deg;C

</span>

<p className="info">

{weatherData.weather.main.temp}&deg;C

</p>

<p>

Feels like: {weatherData.weather.main.feels\_like}&deg;C

</p>

<p className="info">

{\_.capitalize(weatherData.weather.weather[0].description)}

</p>

<p className="info">

Humidity: {weatherData.weather.main.humidity}

</p>

</>

)}

{error && <p>{error}</p>}

</div>

</div>

<div className="box2" style={{ background: 'none' }}>

<Time />

<div className="forecasts">

{weatherData && weatherData.forecasts ? (() => {

const forecastElements = [];

for (let i = 8; i < weatherData.forecasts.length; i = i + 8) {

const forecast = weatherData.forecasts[i];

let nextDate = new Date(forecast.dt \* 1000);

forecastElements.push(

<div className="cards" key={i}>

<p id="dates">{nextDate.toLocaleString('en-US', {

weekday: "short", day: "2-digit",

month: "short"

})}</p>

<p id="foreTemp">{forecast.main.temp}&deg;C</p>

<p id="foreDesc">{\_.capitalize(forecast.weather[0].description)}</p>

<p id="foreHumid">Humidity: {forecast.main.humidity}</p>

</div>

);

}

return forecastElements.length > 0 ? forecastElements : <p>No forecasts available</p>;

})() : <p>LOADING...</p>}

</div>

</div>

</div>

</>

);

}

function SearchBar() {

const [searchInput, setSearchInput] = useState('');

async function handleSearch(e) {

e.preventDefault();

try {

const response = await axios.post(`http://127.0.0.1:3001/api/search`, {

search: \_.capitalize(searchInput)

});

const data = response.data;

setWeatherData(data);

setError(null);

}

catch (err) {

console.log(err);

setError(err.data);

setWeatherData(null);

}

};

return (

<>

<div className="searchform">

<form onSubmit={handleSearch}>

<input

className="searchbar"

type="text"

placeholder="Enter to search a city"

value={searchInput}

onChange={(e) => setSearchInput(e.target.value)}

autoComplete="off"

autoCapitalize="on"

/>

<button className="searchbar" type="submit">Search</button>

</form>

</div>

</>

)

}

export default App

//Time.jsx

import { useEffect, useState } from 'react'

function Time(){

const [currentTime, setCurrentTime] = useState('');

const updateTime = () => {

const longtime = new Date();

setCurrentTime(longtime.toLocaleString('en-US', { hour12: true, hour: "2-digit", minute: "2-digit", second: "2-digit"}));

};

useEffect(() => {

updateTime();

const intervalId = setInterval(updateTime, 1000);

return () => clearInterval(intervalId);

}, []);

return (

<span id="time">{currentTime}</span>

);

};

export default Time;

//Date.jsx

import { useEffect, useState } from 'react'

function DateComponent(){

const [currentDate, setCurrentDate] = useState('');

const updateDate = () => {

const dateObj = new Date();

setCurrentDate(dateObj.toLocaleDateString("en-US", { weekday: "long", day: "2-digit", month: "long", year: "numeric"}));

};

useEffect(() => {

updateDate();

}, []);

return (

<span id="date">{currentDate}</span>

);

};

export default DateComponent;

**5. Snapshots**

