Task 3: Weather Forecast Website

A weather forecast website is a web application that provides users with real-time weather data for their location. To complete this task, you'll need to:

Use JavaScript to create interactive client-side functionality, such as fetching weather data and updating the UI

Set up a server-side API using Node.js to fetch weather data from a third-party API

Use ReactJS to build a dynamic and interactive user interface that HTML

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Weather Forecast</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<header>

<h1>Weather Forecast</h1>

<form id="location-form">

<input type="text" id="location-input" placeholder="Enter location">

<button type="submit">Get Weather</button>

</form>

</header>

<main id="weather-container">

<!-- Weather data will be inserted here -->

</main>

<script src="app.js"></script>

</body>

</html>

CSS

JS

document.getElementById('location-form').addEventListener('submit', function(e) {

e.preventDefault();

let locationInput = document.getElementById('location-input').value;

fetchWeather(locationInput);

});

function fetchWeather(location) {

// Replace with your API key and API endpoint

let apiKey = 'your\_api\_key\_here';

let endpoint = `https://api.weatherapi.com/v1/forecast.json?key=${apiKey}&q=${location}&days=5`;

fetch(endpoint)

.then(response => response.json())

.then(data => {

displayWeather(data);

})

.catch(error => console.error('Error fetching weather:', error));

}

function displayWeather(data) {

let weatherInfo = document.getElementById('weather-info');

weatherInfo.innerHTML = ''; // Clear previous data

// Display current weather

let currentWeather = `

<div>

<h3>Today</h3>

<p>Condition: ${data.current.condition.text}</p>

<p>Temperature: ${data.current.temp\_c}°C</p>

</div>

`;

weatherInfo.innerHTML += currentWeather;

// Display forecast for next 5 days

let forecast = data.forecast.forecastday;

forecast.forEach(day => {

let date = new Date(day.date);

let dayOfWeek = date.toLocaleDateString('en-US', { weekday: 'long' });

let forecastItem = `

<div>

<h3>${dayOfWeek}</h3>

<p>Condition: ${day.day.condition.text}</p>

<p>Max Temp: ${day.day.maxtemp\_c}°C</p>

<p>Min Temp: ${day.day.mintemp\_c}°C</p>

</div>

`;

weatherInfo.innerHTML += forecastItem;

});

}