Major Academor Weather project

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      margin: 0;
      display: flex;
      align-items: center;
      justify-content: center;
      height: 100vh;
      width: 100vw;
    }

    .container {
      text-align: center;
      background-color: rgba(148, 150, 148, 0.7);
      padding: 20px;
      border-radius: 10px;
    }

    input {
      padding: 8px;
    }

    button {
```

```
    padding: 8px 16px;
    margin-top: 8px;
    cursor: pointer;
  }

  #currentWeather {
    margin-top: 20px;
  }

  #forecast {
    margin-top: 20px;
    display: flex;
    justify-content: space-around;
  }

  .day {
    flex: 1;
    text-align: center;
    margin: 10px;
    padding: 10px;
    border: 1px solid #ffff;
    border-radius: 5px;
  }

  canvas {
    max-width: 100%;
    width: 300px; /* Adjust the width as needed */
    height: auto;
  }
</style>
<title>Weather Forecast App</title>
```

```
</head>
<body>
  <div class="container">
    <label for="cityInput">Enter City:</label>
    <input type="text" id="cityInput" placeholder="Type city name" value="Delhi">
    <button onclick="getWeather()">Get Weather</button>
    <div id="currentWeather"></div>
    <div id="forecast"></div>
  </div>
  <script>
    const apiKey = '5eedb53da4e31c68c045887c10d842be';
    const defaultCity = 'Delhi';
    const apiUrl = 'https://api.openweathermap.org/data/2.5/forecast';

    async function getWeather() {
      const cityInput = document.getElementById('cityInput');
      const cityName = cityInput.value || defaultCity;

      try {
        const response = await axios.get(${apiUrl}?q=${cityName}&appid=${apiKey}&units=metric);
        const data = response.data;

        if (response.status === 200) {
          displayCurrentWeather(data);
          displayWeatherForecast(data);
        } else {
          console.error('Error fetching weather data:', data?.message || 'Unknown error');
        }
      } catch (error) {
        console.error('Error fetching weather data:', error.message);
      }
```

```
}

function displayCurrentWeather(data) {
  const currentWeatherDiv = document.getElementById('currentWeather');
  currentWeatherDiv.innerHTML = `
    <h2>Current Weather in ${data.city.name}</h2>
    <p>Temperature: ${data.list[0].main.temp}°C</p>
    <p>Humidity: ${data.list[0].main.humidity}%</p>
    <p>Wind Speed: ${data.list[0].wind.speed} m/s</p>
    <p>Weather Condition: ${data.list[0].weather[0].description}</p>
  `;
}

function displayWeatherForecast(data) {
  const forecastDiv = document.getElementById('forecast');
  forecastDiv.innerHTML = '';

  for (let i = 0; i < data.list.length && i < 6 * 8; i += 8) {
    const dayData = data.list[i];
    const date = new Date(dayData.dt * 1000);
    const dayDiv = document.createElement('div');
    dayDiv.className = 'day';

    let chartCanvas = document.createElement('canvas');
    chartCanvas.id = weatherChart${i};
    chartCanvas.width = 300;
    chartCanvas.height = 200;

    dayDiv.innerHTML = `
      <p>${date.toDateString()}</p>
      <p>Temperature: ${dayData.main.temp}°C</p>
```

```
    <p>Humidity: ${dayData.main.humidity}%</p>
    <p>Weather Condition: ${dayData.weather[0].description}</p>
   `;

   dayDiv.appendChild(chartCanvas);
   forecastDiv.appendChild(dayDiv);

   drawCombinedChart(weatherChart${i}, dayData.main.temp, dayData.main.humidity);
  }
 }

function drawCombinedChart(canvasId, tempData, humidityData) {
  const ctx = document.getElementById(canvasId).getContext('2d');

  new Chart(ctx, {
   type: 'bar',
   data: {
    labels: ['Temperature', 'Humidity'],
    datasets: [{
     label: 'Temperature (°C)',
     data: [tempData],
     backgroundColor: 'rgba(255, 0, 0, 0.5)',
     borderColor: 'rgba(255, 0, 0, 1)',
     borderWidth: 2,
    }, {
     label: 'Humidity (%)',
     data: [humidityData],
     backgroundColor: 'rgba(0, 0, 255, 0.5)',
     borderColor: 'rgba(0, 0, 255, 1)',
     borderWidth: 2,
    }]
```

```
      },
      options: {
        scales: {
          y: {
            beginAtZero: true
          }
        }
      }
    });
  }

  getWeather();
 </script>
</body>
</html>
```