Elbette, 5. ve 6. maddeleri daha ayrıntılı bir şekilde açıklayalım:

\*\*5. Hava Durumu Uygulaması:\*\*

Hava durumu uygulaması yapmak, API'leri kullanma ve dış kaynaklardan veri çekme konusunda pratik yapmanıza yardımcı olabilir. İşte adım adım bir hava durumu uygulaması oluşturmanıza yardımcı olabilecek adımlar:

\*\*Adım 1: API Seçimi\*\*

Hava durumu verilerini sağlayan bir API seçmeniz gerekecek. OpenWeatherMap, WeatherAPI ve AccuWeather gibi ücretsiz veya ücretli seçenekler arasından birini seçebilirsiniz. API'yi kullanmak için kaydolun ve API anahtarınızı alın.

\*\*Adım 2: HTML ve CSS Tasarımı\*\*

Kullanıcı arayüzünü tasarlamak için HTML ve CSS kullanın. Bir şehir seçme veya arama kutusu, hava durumu bilgilerini görüntülemek için bir alan ve hava durumu simgesi gibi bileşenleri içeren basit bir web sayfası oluşturun.

\*\*Adım 3: JavaScript Programlaması\*\*

JavaScript'i kullanarak API ile iletişim kurun ve hava durumu bilgilerini çekin. Örneğin, JavaScript Fetch API'sini kullanabilirsiniz:

```javascript

const apiKey = 'YOUR\_API\_KEY';

const apiUrl = `https://api.openweathermap.org/data/2.5/weather`;

function getWeather(city) {

fetch(`${apiUrl}?q=${city}&appid=${apiKey}`)

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

.then((data) => {

// Hava durumu verilerini işleyin ve ekranda gösterin

})

.catch((error) => {

console.error('Hava durumu bilgisi alınamadı: ', error);

});

}

```

\*\*Adım 4: Sonuçları Gösterme\*\*

API'den gelen hava durumu verilerini işleyin ve kullanıcı arayüzünde gösterin. Örneğin, sıcaklık, nem oranı ve hava durumu durumu gibi bilgileri görüntüleyebilirsiniz.

\*\*6. Web Sayfası İnteraktiflik:\*\*

Bir web sayfasına interaktif öğeler eklemek, kullanıcı deneyimini geliştirmek için önemlidir. İşte bu konuda örnekler:

\*\*Örnek 1: Düğmeler ve Olaylar\*\*

HTML'de bir düğme ekleyin ve JavaScript kullanarak düğmeye bir tıklama olayı ekleyin. Bu düğmeye tıklandığında bir metin veya işlevi değiştirmek gibi basit bir interaktivite örneği olabilir:

```html

<!DOCTYPE html>

<html>

<head>

<title>Web Sayfası İnteraktiflik</title>

</head>

<body>

<button id="myButton">Tıkla</button>

<p id="message">Bu bir örnek metin.</p>

<script>

const button = document.getElementById('myButton');

const message = document.getElementById('message');

button.addEventListener('click', () => {

message.textContent = 'Düğmeye tıklandı!';

});

</script>

</body>

</html>

```

\*\*Örnek 2: Form İşleme\*\*

Bir form ekleyerek kullanıcıdan veri alın ve bu veriyi işleyin. Örneğin, bir kullanıcının adını ve soyadını girmesini isteyen ve bunları bir sonuç sayfasında görüntüleyen bir form oluşturabilirsiniz:

```html

<!DOCTYPE html>

<html>

<head>

<title>Form İşleme</title>

</head>

<body>

<form id="myForm">

<label for="firstName">Ad:</label>

<input type="text" id="firstName" name="firstName">

<br>

<label for="lastName">Soyad:</label>

<input type="text" id="lastName" name="lastName">

<br>

<input type="submit" value="Gönder">

</form>

<div id="result"></div>

<script>

const form = document.getElementById('myForm');

const result = document.getElementById('result');

form.addEventListener('submit', (e) => {

e.preventDefault();

const firstName = document.getElementById('firstName').value;

const lastName = document.getElementById('lastName').value;

result.textContent = `Merhaba, ${firstName} ${lastName}!`;

});

</script>

</body>

</html>

```

Bu projeler, JavaScript'i kullanarak temel interaktiflik ve veri alışverişi konularında deneyim kazanmanıza yardımcı olacaktır. Her iki projeyi de geliştirerek daha karmaşık interaktif uygulamalar yapabilirsiniz.