

Proje Raporu: Hava Durumu Sitesi

1. Amacı

Bu projenin amacı, kullanıcılara anlık hava durumu bilgilerini sunan bir web uygulaması geliştirmektir. Kullanıcılar, arama kutusu yardımıyla belirli bir şehri arayarak o şehrin hava durumu verilerine ulaşabilecek ve favorilerine ekleyip sonradan aramaya gerek kalmadan görüntüleyebilecektir. Ayrıca sayfada geliştirici hakkında bilgi bölümü de yer almaktadır. Proje, kullanıcılara sade bir arayüz sunarak hava durumu bilgisine hızlı bir şekilde erişmelerini sağlamayı hedeflemektedir.

2. Kullanılan Teknolojiler

- **HTML:** Web sayfası yapısal olarak oluşturmak için kullanıldı.
- **CSS:** Sayfanın görsel tasarımını yapmak için kullanıldı. Sayfa düzeni, renkler, ve ikonların yerleşimi gibi öğeler CSS ile ayarlandı.
- **JavaScript:** Sayfadaki arama yapma, favorileri gösterme gibi işlevleri gerçekleştirmek amacıyla kullanıldı

3. Projenin Genel Yapısı

HTML

HTML dosyasında iki ana bölüm bulunmaktadır:

- **Üst Bar:** Sayfanın en üstünde bir navigasyon çubuğu ve arama kutusu vardır. Bu bölüm, kullanıcıların sayfada gezinmesini ve arama yapmasını sağlar.
- **Alt Bar:** Sayfanın alt kısmında hava durumu bilgileri yer alır. About sayfasında ise isim, soy isim, eposta gibi bilgiler bulunur.

```
<div class="ustbar">
  <a id="home" class="active1" href="index.php">
    <svg fill="currentcolor" xmlns="http://www.w3.org/2000/svg" id="Layer_1" data-name="Layer 1" viewBox="0 0 24 24" width="16" height="16">
      Hava Durumu</a>
    <input type="search" id="aramakutusu" onkeydown="aramaYapma(this)" placeholder="Aramak için yazın..">
    <button type="button" id="aramabuton" onclick="aramaYapma(this)">
      <svg width="25px" height="25px" viewBox="0 0 20 20" xmlns="http://www.w3.org/2000/svg" fill="none">
        <path fill="#e4e4e4" fill-rule="evenodd" d="M4 9a5 5 0 110 0a5 5 0 014 9zm5-7a7 7 0 104.2 12.6.999 0 00.093.107l3 3a1 1 0 001 1" />
      </svg>
    </button>
    <a id="about" href="about.php">
      <svg fill="currentcolor" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" width="16" height="16"><path d="M12,24A12,12,0,1,0,12,0A12,12,0,1,0,12,24" />
    </a>
</div>
```

Bu bölümdeki ana elemanlar şunlardır:

- **Hava Durumu Butonu:** Sayfanın ana başlığı ve anasayfaya dönmeyi sağlayan buton.
- **Arama Kutusu:** Kullanıcıların arama yapmasına olanak tanıyan bir input kutusu. onkeydown ve onclick olayları ile arama JavaScript fonksiyonlarına bağlanmıştır.
- **Hakkında Butonu:** Sayfa içindeki bir buton ile geliştirici bilgilerine erişim sağlanmıştır.

CSS

CSS dosyası ile sayfanın görünümü ve yerleşimi düzenlenmiştir. Farklı ekran boyutlarında görselliği korumak amaçlanmıştır. Ayrıca butonlar ve arama kutusu gibi elemanlara uygun görsel stil verilmiştir. Örneğin:

```
.ustbar{
  background-color: var(--topbar-color);
  overflow: hidden;
  min-width: 720px;
  width: 100%;
}

.ustbar a{
  color: var(--text-color);
  font-family: var(--font-all);
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 14px;
}

.ustbar a:hover{
  background-color: var(--topbar-hover);
  color: var(--text-color);
  border-radius: 14px;
}

.ustbar a.active1{
  background-color: var(--topbar-active);
  color: var(--text-color);
  border-radius: 14px;
}

.ustbar input[type=search] {
  background-color: var(--topbar-active);
  color: var(--text-color);
}
```

```
.altbar {
  width: 65%;
  height: 580px;
  min-width: 690px;
  border-radius: 14px;
  align-self: center;
  padding: 25px 16px;
  margin-left: auto;
  margin-right: auto;
}

.infobar {
  background-color: var(--bottom-bar);
  width: 90%;
  height: 90%;
  float: left;
  border-radius: 14px;
  padding: 30px 30px;
  position: relative;
  color: var(--text-color);
  font-family: var(--font-all);
  font-size: 25px;
}

.solbar {
  background-color: var(--bottom-bar);
  width: 56%;
  height: 95%;
  float: left;
  border-radius: 14px;
  padding: 16px 16px;
}
```

Bu stiller ile kullanıcı arayüzü şu şekilde düzenlenmiştir:

- Arka plan renkleri ve yazı renkleri ayarlanarak kontrast sağlanmıştır.
- Sayfadaki elementlerin pozisyonları, büyüklükleri ayarlanarak kullanıcı arayüzünün okunabilirliği artırılmıştır.

JavaScript

JavaScript dosyasında, arama fonksiyonu, hava durumunun API kullanılarak sayfaya verilmesi ve kullanıcı etkileşimleri yönetilmiştir. Sayfa ilk açıldığında fetch fonksiyonu kullanılarak favorilerde bulunan şehirlerin hava durumları ekrana verilir. Sayfa içinde arama kutusuna bir kelime yazıldığında veya arama butonuna tıklanıldığında aramaYapma fonksiyonu tetiklenir. Bu fonksiyon, arama sonucunda doğru hava durumu verisini alarak ekrana verir.

```
async function favoriHavaDurum(sehir1) : Promise<...> {
  deger = geocodeURL(geocodeApiKey, sehir1);
  deger2 = await degerAl(deger);
  havadurum1 = deger2[0].wxPhraseLong;
  derece1 = Math.round( X: (5/9)*((await deger2[0].temperature)-32))+"°C";
  image1 = havaDurumIkon(deger2[0].iconCode);
  image = "<img src='icons/image'>".replace( searchValue: "image", image1);
  sehir = deger2[1];

  const favoriHavaDurumu : {...} = {"sehir": sehir, "havadurum": havadurum1, "derece": derece1, "image": image, "object": deger2[0]};

  return favoriHavaDurumu;
}
```

```

async function mainHavaDurum(sehir, index) : Promise<void> {
    durum = favoriHavaObject[index];

    havadurum = durum.wxPhraseLong;
    derece = Math.round( (x: (5/9)*((await durum.temperature)-32))+"°C";
    image = havaDurumIkon(durum.iconCode);
    airpressure = durum.pressureMeanSeaLevel+" mBar";
    humidity = durum.relativeHumidity+"%";
    feeltemp = Math.round( (x: (5/9)*((await durum.temperatureFeelsLike)-32))+"°C";
    uv = durum.uvDescription+"/ "+durum.uvIndex;
    visibility = durum.visibility+" km";
    windspeed = durum.windSpeed+" km/s";
    winddirection = durum.windDirectionCardinal.slice(-2);

    document.getElementById( elementId: "solbar-image").innerHTML = "<img src='icons/image'>".replace( searchValue: "image", image);
    document.getElementById( elementId: "solbar-sehir").innerHTML = sehir;
    document.getElementById( elementId: "solbar-havadurumu").innerHTML = havadurum;
    document.getElementById( elementId: "solbar-derece").innerHTML= derece;
    document.getElementById( elementId: "solbar-airpressure-value").innerHTML= airpressure;
    document.getElementById( elementId: "solbar-nem-value").innerHTML = humidity;
    document.getElementById( elementId: "solbar-hissicak-value").innerHTML = feeltemp;
    document.getElementById( elementId: "solbar-uv-value").innerHTML = uv;
    document.getElementById( elementId: "solbar-vis-value").innerHTML = visibility;
    document.getElementById( elementId: "solbar-wind-value").innerHTML = windspeed;
    document.getElementById( elementId: "solbar-winddir-value").innerHTML = winddirection;
}

```

```

function aramaYapma(element) : void {
    if ((event.key === 'Enter') || (element.nodeName === "BUTTON")){
        aramaKapatma();
        const arkaplan : HTMLDivElement = document.createElement( tagName: "div");
        arkaplan.setAttribute( qualifiedName: "id", value: "transparent_background");
        arkaplan.setAttribute( qualifiedName: "onclick", value: "aramaKapatma()");
        document.getElementById( elementId: "overlayid").appendChild(arkaplan);

        aramaGoruntule();
    }
}

```

```

async function aramaGoruntule() : Promise<void> {
    let searchResult : {...} = await favoriHavaDurum(document.querySelector( selectors: '#aramakutusu').value);

    const arama : HTMLDivElement = document.createElement( tagName: "div");
    arama.setAttribute( qualifiedName: "id", value: "arama_sonuc");

    const arama_image : HTMLDivElement = document.createElement( tagName: "div");
    arama_image.innerHTML = searchResult.image;
    arama_image.setAttribute( qualifiedName: "id", value: "arama_image");
    arama.appendChild(arama_image);

    const arama_sehir : HTMLSpanElement = document.createElement( tagName: "span");
    arama_sehir.innerHTML = searchResult.sehir;
    arama_sehir.setAttribute( qualifiedName: "id", value: "arama_sehir");
    arama.appendChild(arama_sehir);

    const arama_derece : HTMLSpanElement = document.createElement( tagName: "span");
    arama_derece.innerHTML = searchResult.derece;
    arama_derece.setAttribute( qualifiedName: "id", value: "arama_derece");
    arama.appendChild(arama_derece);

    const arama_havadurumu : HTMLSpanElement = document.createElement( tagName: "span");
    arama_havadurumu.innerHTML = searchResult.havadurum;
    arama_havadurumu.setAttribute( qualifiedName: "id", value: "arama_havadurumu");
    arama.appendChild(arama_havadurumu);

    const arama_form : HTMLFormElement = document.createElement( tagName: "form");
    arama_form.setAttribute( qualifiedName: "action", value: "index.php");
    arama_form.setAttribute( qualifiedName: "method", value: "post");
}

```

4. Projenin Özellikleri

- **Arayüzü:**
 - Sade ve kullanıcı dostu bir tasarım.
 - Sağ tarafta favori şehirler ve hava durumu bilgileri.
 - Sol tarafta seçili şehrin hava durumu hakkında ayrıntılı bilgi.
 - Hava durumu arama kutusu ve butonu.
- **İşlevsellik:**
 - JavaScript ile arama işlemi gerçekleştirilmekte.
 - Kullanıcıların girdiği verilere ve kayıtlı şehirlere göre hava durumu sonuçları döndürülmekte.
 - PHP kullanılarak favorilere ekleme çıkartma işlemi yapılabilmekte.
- **Geliştirici Bilgileri:** Sayfanın about kısmında, proje geliştiricisinin ad, soyad, bölüm bilgisi ve e-posta adresi gibi iletişim bilgileri yer almaktadır.

5. Proje Adımları

1. **HTML ve CSS ile Temel Tasarım:** Sayfa yapısının oluşturulması, arama kutusu ve üst/alt barların yerleştirilmesi ile başladı.
2. **JavaScript ile Dinamik Fonksiyonların Eklenmesi:** Arama butonunun ve kutusunun işlevi ve projenin ana işlevi olan API kullanılarak hava durumu verilerinin alınması için JavaScript kodları eklendi.
3. **PHP ile JSON Dosyası Düzenleme:** PHP kullanarak favorilerin bir JSON dosyasına eklenip çıkartılabilmesi sağlandı.
4. **Geliştirici Bilgilerinin Eklenmesi:** About sayfasında geliştiriciye ait bilgilerin yerleştirilmesi.
5. **Tasarım Düzenlemeleri:** Sayfanın farklı ekran boyutlarında düzgün görüntülenmesi için CSS düzenlemeleri.

6. Sonuç

Proje, temel bir hava durumu uygulaması için gereken arayüzü ve işlevselliği sağlamaktadır. Kullanıcılar, arama yaparak hava durumu bilgisini sorgulayabilir, elde ettikleri sonuçları favorilerine ekleyerek kaydedebilir ve geliştirici hakkında bilgilere ulaşabilirler. Proje şu an için temel düzeyde çalışmakta olup, ilerleyen aşamalarda farklı hava durumu sitelerinin API'ları da eklenip karşılaştırılabilir. Ayrıca daha fazla şehir ve bölgeye özel arama yapılabilmesi için gelişmiş filtreleme özellikleri eklenebilir.

7. İyileştirmeler

- **Mobil Uyumluluk:** CSS düzenlemeleriyle mobil cihazlar için daha optimize bir kullanıcı deneyimi sunulması.

8. Özet

Bu proje, temel HTML, CSS ve JavaScript bilgileri kullanarak bir hava durumu uygulaması oluşturmanın bir örneğidir. Kullanıcı dostu arayüzü ve dinamik fonksiyonları ile başlangıç düzeyinde bir proje olmuştur.

Video Linki: <https://drive.google.com/file/d/1mPiGws4rGOLp2JTlnr7GN8YzbKUMuSSF/view?usp=sharing>