



# Belirli Şehirlerin Anlık Hava Durumu Koşullarının Hesaplama Ve Gösterimi

Yazılım Mühendisliği Ana Bilim Dalı  
Dönem Projesi

Anıl Özdağ  
0009-0000-6774-4360

Proje Danışmanı: Dr. Öğretim Üyesi Emre Şatır

Haziran, 2024

T.C.  
İZMİR KATİP ÇELEBİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
YAZILIM MÜHENDİSLİĞİ ANABİLİM DALI

# Belirli Şehirlerin Anlık Hava Durumu Koşullarının Hesaplama Ve Gösterimi

Yazılım Mühendisliği Ana Bilim Dalı  
Dönem Projesi

Anıl Özdağ  
0009-0000-6774-4360

Proje Danışmanı: Dr. Öğretim Üyesi Emre Şatır

Haziran, 2024

# YEMİN METNİ

Yüksek Lisans Projesi olarak sunduğum " Belirli Şehirlerin Anlık Hava Durumu Koşullarının Hesaplama Ve Gösterimi" adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

24/06/2024

Anıl ÖZDAĞ

# DÖNEM PROJESİ ONAY FORMU

*Bu sayfaya 8E formu word ortamında doldurulduktan ve ilgili imzalar alındıktan sonra yerleştirecektir.*

*Sıralama için bu sayfaya çıkarmayın.*

# Belirli Şehirlerin Anlık Hava Durumu Koşullarının Hesaplama Ve Gösterimi

## ÖZ

Bu çalışmada, kullanıcı dostu bir arayüz aracılığıyla hava durumu bilgilerini sunan bir Windows Forms uygulaması geliştirilmiştir. Uygulama, C# programlama dili ve .NET Framework kullanılarak MVC mimarisi ile tasarlanmıştır. Kullanıcılar, çeşitli şehirler için güncel hava durumu bilgilerine erişebilir ve 1 haftalık hava durumu verilerini sorgulayabilirler. Uygulama, dinamik veri işleme ve grafiksel veri görselleştirme teknikleri kullanarak, kullanıcılara anlaşılır ve etkileşimli bir deneyim sunmaktadır. Güvenlik özellikleri, gelişmiş şifreleme ve veri koruma tekniklerini içerir. Ayrıca, uygulama asenkron programlama tekniklerini ve verimli veri sorgulama yöntemlerini kullanarak performans optimizasyonu sağlamaktadır.

**Anahtar Sözcükler:** Windows Forms, C#, .NET Framework, hava durumu, veri görselleştirme, MVC mimarisi, asenkron programlama.

# Calculation And Display Of Instant Weather Conditions Of Specific Cities

## Abstract

This study develops a Windows Forms application that presents weather information through a user-friendly interface. The application is designed using C# and the .NET Framework, structured around the MVC architecture. Users can access current weather information for various cities and query weather data for a week. The application employs dynamic data processing and graphical data visualization techniques to provide an understandable and interactive experience. Security features include advanced encryption and data protection techniques. Additionally, the application enhances performance through asynchronous programming techniques and efficient data querying methods.

**Keywords:** Windows Forms, C#, .NET Framework, weather information, data visualization, MVC architecture, asynchronous programming.

# Teşekkür

Bu tez çalışması, İzmir Katip Çelebi Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği Anabilim Dalı'nda yüksek lisans öğrenimim sırasında gerçekleştirilmiştir. Bu çalışmanın amacı, belirli şehirlerin anlık hava durumu koşullarını hesaplamak ve görselleştirmek üzere bir Windows Forms uygulaması geliştirmektir. Uygulama, C# programlama dili ve .NET Framework kullanılarak Model-View-Controller (MVC) mimarisi ile tasarlanmıştır.

Bu çalışmanın gerçekleştirilmesinde katkı sağlayan kişi ve kurumlara teşekkür etmek isterim. Öncelikle, tez danışmanım Dr. Öğretim Üyesi Emre Şatır'a, çalışmanın her aşamasında gösterdiği rehberlik, destek ve teşvik için sonsuz teşekkürlerimi sunarım.

Ayrıca, araştırma sürecinde bilgi ve tecrübelerini benimle paylaşan tüm akademisyenlere ve arkadaşlarıma teşekkürü bir borç bilirim. Çalışmam sırasında bana sürekli destek olan ve sabır gösteren aileme en derin şükranlarımı sunarım.

Bu tez çalışmasının, hava durumu verilerinin hesaplanması ve görselleştirilmesi alanında yeni çalışmalar için bir temel oluşturmasını ve katkı sağlamasını umuyorum.

Anıl ÖZDAĞ

İzmir-2024

# İçindekiler

Öz.....	i
Abstract.....	ii
Teşekkür.....	iii
Şekiller Listesi.....	vi
<b>1 Giriş.....</b>	<b>1</b>
<b>2 Literatür Taraması.....</b>	<b>2</b>
2.1 Hava Durumu Uygulamalarının Gelişimi.....	2
2.2 Veri Kaynakları Ve Toplama Yöntemleri.....	2
2.3 Görselleştirme Teknikleri.....	3
2.4 Kullanıcı Arayüzü Tasarımı.....	3
2.5 Performans Ve Optimizasyon.....	3
<b>3 Projede Kullanılan Teknolojiler.....</b>	<b>4</b>
3.1 Mvc Mimarisi Ve Avantajları.....	4
3.2 Geliştirme Sürecinde Karşılaşılan Zorluklar Ve Çözümler.....	5
3.2.1 Veri Entegrasyonu ve İşleme.....	5
3.2.2 Kullanıcı Arayüzü Tasarımı.....	5
3.2.3 Performans Optimizasyonu.....	6
3.3 Kullanılan Kütüphaneler Ve Araçlar.....	6
3.4 Veri İşleme Ve Görselleştirme.....	7
3.5 Güvenlik Ve Performans.....	7
<b>4 Yöntemler.....</b>	<b>9</b>
4.1 Yazılım Geliştirme Araçları Ve Ortamları.....	9
4.2 Veri Toplama Ve İşleme.....	9



4.3 Kullanıcı Arayüzü Tasarımı Ve Geliştirme .....	10
4.4 Test, Doğrulama, Performans Ve Optimizasyon.....	12
4.5 Görselleştirme Ve Sonuçlar Sunumu .....	12
<b>5 Uygulama Mimarisi ve Yönetimi .....</b>	<b>14</b>
<b>6 Bulgular .....</b>	<b>17</b>
6.1 Görselleştirme Ve Arayüz Tasarımı.....	17
6.2 Veri İşleme Ve Performans .....	17
6.3 Kullanıcı Deneyimi Ve İnteraktivite .....	18
6.4 Teknik Yapı Ve Güvenlik Özellikleri .....	18
<b>7 Tartışma.....</b>	<b>19</b>
7.1 Test Süreçleri Ve Elde Edilen Sonuçlar .....	19
7.2 Genel Tartışma Ve Gelecek Çalışmalar .....	19
<b>8 Sonuç ve Öneriler.....</b>	<b>21</b>
8.1 Sonuçlar.....	21
8.2 Uygulama Ve Öneriler .....	21
<b>Kaynaklar .....</b>	<b>23</b>

# Şekiller Listesi

Şekil 4.1	Uygulamanın ana sayfası .....	<b>10</b>
Şekil 4.2	Şehir arama fonksiyonu.....	<b>11</b>
Şekil 4.3	Hata ekranı ekran görüntüsü .....	<b>11</b>
Şekil 4.4	Haftalık hava durumu tahmini.....	<b>13</b>

# Bölüm 1

## Giriş

Günümüzde, hızla değişen hava koşullarına karşı bilinçli olmak hem günlük yaşamı planlamak hem de olası risklerden korunmak açısından büyük önem taşımaktadır. Bu bağlamda, güncel ve tarihsel hava durumu verilerine hızlı ve kolay erişim sağlayan teknolojik çözümlere olan ihtiyaç artmaktadır. Geliştirilen bu Windows Forms uygulaması, kullanıcı dostu bir arayüz aracılığıyla hava durumu bilgilerini sunarak bu ihtiyaca yanıt vermektedir. Uygulama, C# ve .NET Framework kullanılarak geliştirilmiş olup, MVC mimarisi ile yapılandırılmıştır. Kullanıcılar, farklı şehirler için hava durumu bilgilerini anında sorgulayabilir ve 1 haftalık hava durumu verilerine ulaşabilirler.

Uygulamanın temel amacı, hava durumu verilerini efektif bir şekilde işleyerek, kullanıcıların bu bilgilere hızlı ve etkili bir şekilde erişimini sağlamaktır. Böylece, kullanıcılar günlük veya 1 haftalık planlarını hava durumu bilgilerine göre şekillendirebilirler. Uygulama, ayrıca gelişmiş veri görselleştirme teknikleri kullanarak, hava durumu bilgilerini anlaşılır ve etkileşimli grafiklerle sunmaktadır.

Bu çalışmanın kapsamı, uygulamanın tasarımı, geliştirilmesi ve kullanıcı arayüzü üzerine yoğunlaşmaktadır. Çalışma, hava durumu verilerinin nasıl işlendiği, kullanıcı arayüzünün nasıl tasarlandığı ve uygulamanın son kullanıcılar tarafından nasıl değerlendirildiği üzerine detaylı bilgiler sunmaktadır.

# Bölüm 2

## Literatür Taraması

### 2.1 Hava Durumu Uygulamalarının Gelişimi

Hava durumu tahmin teknolojileri, özellikle son yüzyılda büyük ilerleme kaydetmiştir. Gelişmiş uydu sistemleri ve radar teknolojileri, daha doğru ve zamanında tahminler yapılmasını sağlamaktadır. Yapay zeka ve makine öğrenimi tekniklerinin entegrasyonu, modelleme ve tahmin süreçlerini otomatize etmekte, böylece meteorologlara ve kullanıcılara daha kapsamlı bilgiler sunmaktadır. Örneğin, IBM'in The Weather Company'si, AI destekli hava durumu tahmin sistemi ile yerel tahminlerde önemli iyileştirmeler sağlamıştır.

### 2.2 Veri Kaynakları Ve Toplama Yöntemleri

Hava durumu tahminlerinde kullanılan verilerin büyük bir kısmı uydu ve radar sistemlerinden sağlanmaktadır. Bu sistemler, atmosferin farklı katmanlarını tarayarak geniş bir veri yelpazesi toplar. Örneğin, GOES-R serisi uydular, hava durumu izleme ve tahmin kapasitesini önemli ölçüde artırmıştır. Veri toplama teknolojilerindeki bu ilerlemeler, tahminlerin doğruluk oranını artırırken, hava durumu uyarılarının hızını da maksimize etmektedir.

## 2.3 Grselleřtirme Teknikleri

Hava durumu verilerinin grselleřtirilmesi, bilginin anlařılmasını kolaylařtırır. Geliřmiř grafik ve animasyon teknikleri, kullanıcılara atmosferik olayları  boyutlu olarak gstermektedir. rneęin, Windy ve Ventusky gibi uygulamalar, rzgar ve hava akıřı modellerini dinamik haritalar zerinde gerek zamanlı olarak sunmaktadır. Bu tr grselleřtirmeler, hem eęitici hem de uyarıcı iřlev grerek kullanıcıların hava durumu deęiřikliklerine karřı bilinlenmesine yardımcı olmaktadır.

## 2.4 Kullanıcı Arayz Tasarımı

Etkili bir kullanıcı arayz tasarımı, hava durumu uygulamalarının bařarısında kritik bir rol oynamaktadır. Kullanıcı arařtırmaları, arayz tasarımının kullanıcı memnuniyetini nemli lde etkiledięini gstermektedir. zellikle dokunmatik ekran optimizasyonu ve kiřiselleřtirme zellikleri, kullanıcıların uygulamayla etkileřimini artırmaktadır.

## 2.5 Performans Ve Optimizasyon

Hava durumu uygulamalarında performans optimizasyonu, veri iřleme ve yanıt srelerinin iyileřtirilmesini ierir. eřitli algoritma iyileřtirmeleri ve veri sıkıřtırma teknikleri, bu uygulamaların daha hızlı alıřmasını saęlamaktadır. Ayrıca, bulut biliřim teknolojilerinin kullanımı, byk veri setlerinin iřlenmesinde nemli rol oynamakta ve uygulamaların leklenebilirlięini artırmaktadır.

# Bölüm 3

## Projede Kullanılan Teknolojiler

Bu projede, C# programlama dili ve .NET Framework platformu kullanılmıştır. Bu teknolojiler sayesinde, Windows Forms tabanlı bir kullanıcı arayüzü geliştirilmiştir. Uygulama, Model-View-Controller (MVC) mimarisi ile yapılandırılarak, kodun sürdürülebilirliği ve modülerliği artırılmıştır. MVC mimarisi, uygulamanın iş mantığını kullanıcı arayüzünden ayırarak, daha temiz ve yönetilebilir bir kod yapısı sunar.

### 3.1 Mvc Mimarisi Ve Avantajları

MVC (Model-View-Controller) mimarisi, yazılım geliştirme sürecinde uygulama mantığını üç ana bileşene ayırır: Model, View ve Controller. Bu yapı, projenin daha düzenli ve yönetilebilir olmasını sağlar, ayrıca geliştirme ve bakım süreçlerinde verimliliği artırır.

*Model:* Uygulamanın veri yapısını temsil eder ve iş kurallarını, veri erişimini, ve veri manipülasyonunu yönetir.

*View:* Kullanıcıya bilgi sunumunu sağlayan arayüz bileşenidir. View, kullanıcı etkileşimlerini alır ve kullanıcıya görsel çıktı sunar.

*Controller:* Kullanıcı girişlerini alır, Model'i kullanarak bu verileri işler ve sonuçları View'a ileterek kullanıcının görmesini sağlar.

Bu mimarinin projeye avantajları şunlardır:

*Geliştirme Sürecinin Hızlanması:* MVC mimarisi, geliştiricilerin View ve Controller katmanlarını bağımsız olarak geliştirmesine olanak tanır, bu da ekip çalışmasını ve paralel geliştirme süreçlerini kolaylaştırır.

*Daha İyi Test Edilebilirlik:* MVC, uygulamanın test edilebilirliğini artırır. Model bağımsız olarak test edilebilir, Controller ise kullanıcı arayüzünden bağımsız bir şekilde birim testlerine tabi tutulabilir.

*Esneklik ve Ölçeklenebilirlik:* Controller'ın arayüzden ve iş kurallarından soyutlanması, uygulamanın yeni özelliklerle kolayca genişletilmesini sağlar.

## 3.2 Geliştirme Sürecinde Karşılaşılan Zorluklar Ve Çözümler

### 3.2.1 Veri Entegrasyonu ve İşleme

*Zorluk:* Farklı API kaynaklarından gelen hava durumu verilerinin entegrasyonu sırasında veri uyumsuzlukları yaşandı. Verilerin format ve zaman damgası açısından farklılıklar göstermesi, uygulamanın doğru tahminler yapmasını zorlaştırıyordu.

*Çözüm:* Veri normalleştirme teknikleri kullanılarak, farklı kaynaklardan alınan veriler tek bir standart formata dönüştürüldü. Bu işlem, veri işleme modüllerinin daha etkili çalışmasını sağladı ve veri akışını optimize etti.

### 3.2.2 Kullanıcı Arayüzü Tasarımı

*Zorluk:* Kullanıcıların uygulamayı etkili bir şekilde kullanabilmesi için intuitif bir arayüz tasarımı gerekiyordu. İlk tasarımlarda bazı navigasyon sorunları yaşandı.

*Çözüm:* Yapılan uygulama testleri sonucunda UI/UX tasarımı ile çalışılarak arayüz yeniden tasarlandı. Menüler ve butonlar daha erişilebilir hale getirildi, renk paleti ve tipografi kullanıcı dostu olacak şekilde ayarlandı.

### 3.2.3 Performans Optimizasyonu

*Zorluk:* Uygulamanın veri işleme hızı, büyük veri setleri ile çalışırken yetersiz kaldı. Özellikle veri görselleştirme sırasında yavaşlamalar meydana geldi.

*Çözüm:* Algoritma optimizasyonları yapılarak ve veri sıkıştırma teknikleri uygulanarak işleme süreçleri hızlandırıldı. Ayrıca, bulut tabanlı çözümler kullanılarak işlem gücü artırıldı ve veri depolama kapasitesi genişletildi.

## 3.3 Kullanılan Kütüphaneler Ve Araçlar

**Bunifu UI Framework:** Uygulamanın görsel arayüz bileşenleri için kullanılmıştır. Bu framework, modern ve estetik kullanıcı arayüzleri oluşturulmasını sağlar.

**Newtonsoft.Json:** API'den gelen JSON verilerinin işlenmesi için kullanılmıştır. Bu kütüphane, JSON formatındaki verilerin .NET nesnelere dönüştürülmesini ve tersini kolaylaştırır.

**OpenWeatherMap API:** Uygulama, hava durumu verilerini gerçek zamanlı olarak çekmek için OpenWeatherMap'in API servisini kullanmaktadır.



## 3.4 Veri İşleme Ve Görselleştirme

Uygulama, hava durumu verilerini grafiksel olarak sunmak için çeşitli veri işleme ve görselleştirme tekniklerini kullanmaktadır. Kullanıcılar, sorguladıkları şehrin hava durumu tahminlerini grafik ve çizelgeler aracılığıyla görüntüleyebilir. Bu, kullanıcıların hava durumu trendlerini daha iyi anlamalarını ve gelecek günler için plan yapmalarını sağlar.

## 3.5 Güvenlik Ve Performans

Uygulama, güvenlik ve performans optimizasyonları açısından dikkatlice tasarlanmıştır. Veri iletimi sırasında SSL şifrelemesi kullanılmış ve uygulamanın hızlı ve verimli çalışması için asenkron programlama teknikleri entegre edilmiştir. Bu da, kullanıcı verilerini koruma konusunda yüksek standartlarda güvenlik önlemleri sunmaktadır. Literatürdeki birçok uygulamanın aksine, güçlü veri şifreleme ve güvenli aktarım teknikleri içermektedir.

Uygulamada uygulanan güvenlik önlemleri, kullanıcı verilerinin korunması ve uygulamanın güvenilir bir şekilde çalışmasını sağlamak için hayati öneme sahiptir. Bu güvenlik önlemleri şunları içerir:

**Veri Şifreleme:** Hassas kullanıcı verileri, veri tabanında saklanırken ve internet üzerinden iletilirken şifrelenir. Bu, veri sızıntısı veya izinsiz erişim durumunda bile bilgilerin okunamaz hale gelmesini sağlar.

**Güvenli Oturum Yönetimi:** Oturum bilgileri güvenli bir şekilde yönetilir ve oturum çerezleri, güvenlik ihlallerini önlemek için HttpOnly ve Secure bayrakları ile işaretlenir.

**Güvenlik Duvarı ve Erişim Kontrolleri:** Uygulamaya yapılan istekler, kötü niyetli etkinlikleri filtrelemek ve izinsiz erişimleri engellemek için güvenlik duvarı tarafından kontrol edilir.

Bu güvenlik önlemleri, kullanıcıların kişisel bilgilerini korumanın yanı sıra, uygulamanın bütünlüğünü ve güvenilirliğini de sağlar, böylece kullanıcılar uygulamayı güven içinde kullanabilir.

# Bölüm 4

## Yöntemler

Bu bölümde, Windows Forms tabanlı hava durumu uygulamasının geliştirilmesi süreci ve kullanılan teknikler detaylandırılmaktadır.

### 4.1 Yazılım Geliştirme Araçları Ve Ortamları

Proje, Microsoft'un .NET Framework'ü kullanılarak C# programlama dilinde geliştirilmiştir. Visual Studio IDE, uygulamanın kod yazımı, hata ayıklama ve dağıtımı için kullanılan ana geliştirme ortamıdır. MVC (Model-View-Controller) mimarisi, uygulamanın tasarımında rehberlik eden yapısal bir model olarak benimsenmiştir.

### 4.2 Veri Toplama Ve İşleme

Uygulamada gerçek zamanlı veri sağlayan API'lar tercih edilmiştir. Bu, literatürde sıklıkla rastlanan daha statik veri kaynaklarına kıyasla bir avantaj sağlar. Özellikle, hava durumu uygulamalarında anlık güncellemeler ve tahminler sunabilme yeteneği, kullanıcılar için büyük bir değer yaratmaktadır.

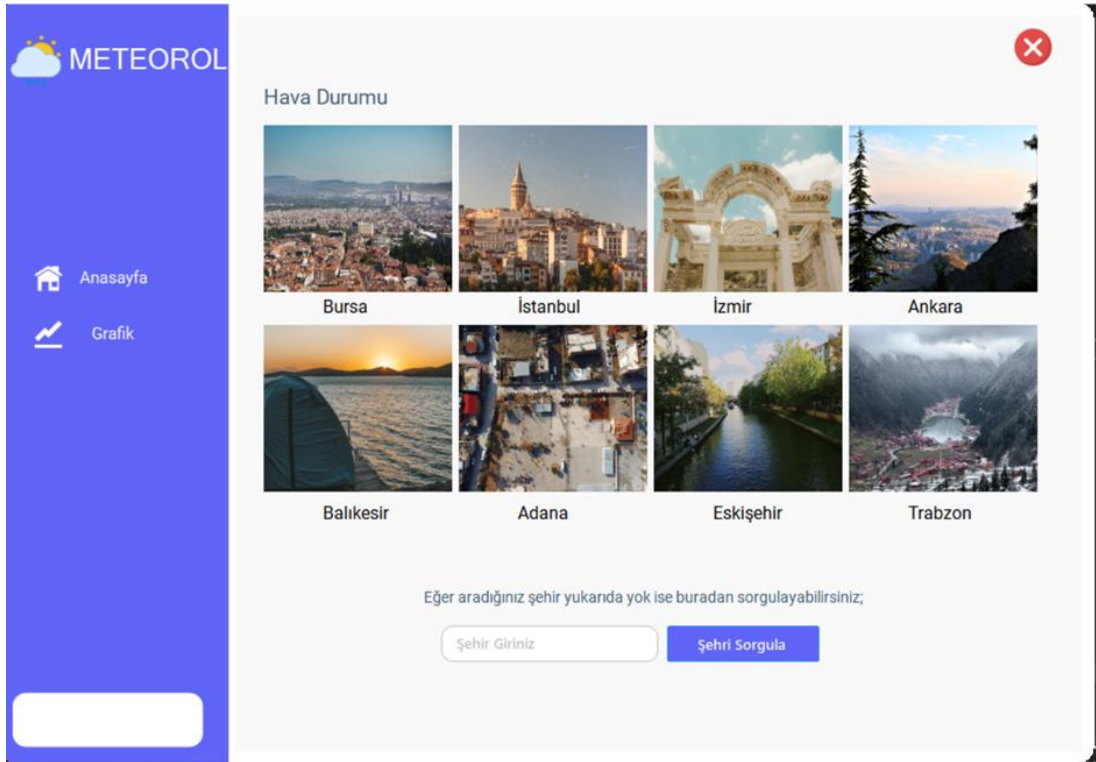
Hava durumu verileri, çeşitli hava durumu API'larından JSON formatında dinamik olarak çekilmiştir. Bu veriler, uygulama içinde işlenip, kullanıcı arayüzünde görselleştirilmeden önce bir dizi işlemde geçirilmiştir. Veri işleme süreci, veri

temizleme, normalizasyon ve ölçeklendirme adımlarını içermektedir. Uygulamanın mimari yapısı, modüler ve katmanlı bir yapı üzerine kurulmuştur.

### 4.3 Kullanıcı Arayüzü Tasarımı Ve Geliştirme

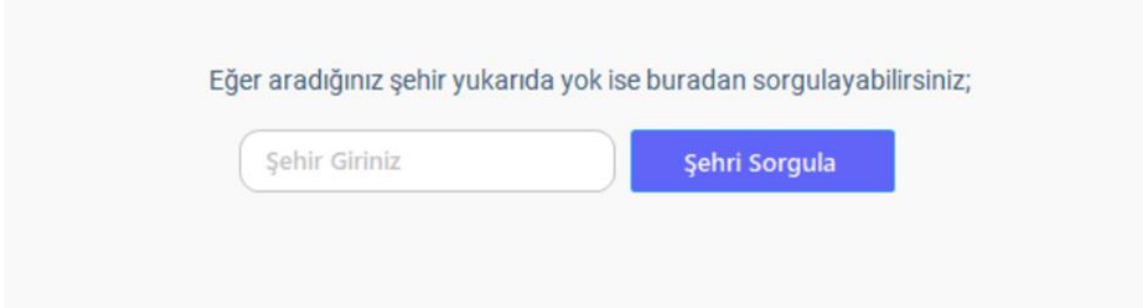
Uygulama kullanıcı dostu bir arayüzle tasarlanmış ve bu, kullanıcıların hava durumu bilgilerine hızlı ve kolay erişimini sağlamaktadır. Literatürdeki çoğu uygulamaların aksine, kişiselleştirme ve kullanıcı deneyimine odaklanarak kullanıcı memnuniyetini artıran bir özelliktir.

Kullanıcı arayüzü, kullanıcı dostu ve etkileşimli bir deneyim sağlamak için tasarlanmıştır. Arayüz elemanları, kullanıcının veriye kolay erişimini ve veriler arasında sezgisel gezinmeyi destekleyecek şekilde düzenlenmiştir.



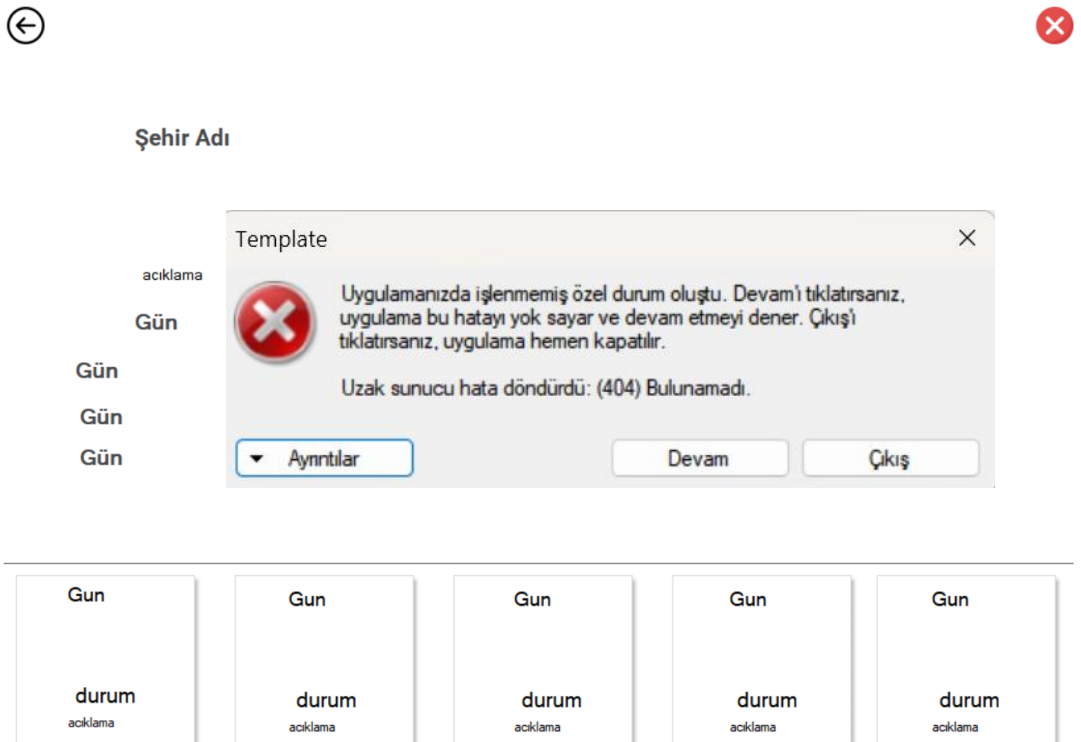
Şekil 4.1: Uygulamanın ana sayfası

Şekil 4.1: Kullanıcılar bu ekran aracılığıyla farklı şehirler için hava durumu bilgilerini görselleştirilebilir bir biçimde görebilir ve detaylı arama yapabilirler. Bu ekran, kullanıcılara sezgisel ve etkileşimli bir deneyim sunmayı amaçlar.



Şekil 4.2: Şehir arama fonksiyonu

Şekil 4.2: Kullanıcıların istedikleri şehrin hava durumu bilgilerine ulaşmalarını sağlayan arama çubuğu. Bu fonksiyon, kullanıcılara hızlı ve doğrudan erişim sağlar ve uygulamanın kullanılabilirliğini artırır.



Şekil 4.3: Hata ekranı ekran görüntüsü

Şekil 4.3: Uygulamanın bir hata durumunda kullanıcıya sunduğu bilgilendirme ekranını göstermektedir. Kullanıcı buradan hatanın kaynağını ve çözüm önerilerini görebilir.

## 4.4 Test, Doğrulama, Performans Ve Optimizasyon

Uygulama, geliştirme sürecinin her aşamasında çeşitli testlerden geçirilmiştir. Birim testleri, her modülün doğru çalıştığını doğrulamak için kullanılmıştır. Entegrasyon testleri, farklı uygulama bileşenlerinin birlikte uyumlu şekilde çalıştığını test etmiştir.

Uygulamada veri işleme ve yanıt sürelerindeki performansa özen gösterilmiştir. Literatürdeki standartın üzerine çıkılmaya çalışılmıştır. Özellikle bulut tabanlı hizmetlerle entegrasyon, uygulamanın büyük veri setlerini hızlı bir şekilde işlemesine olanak tanımıştır.

## 4.5 Görselleştirme Ve Sonuçlar Sunumu

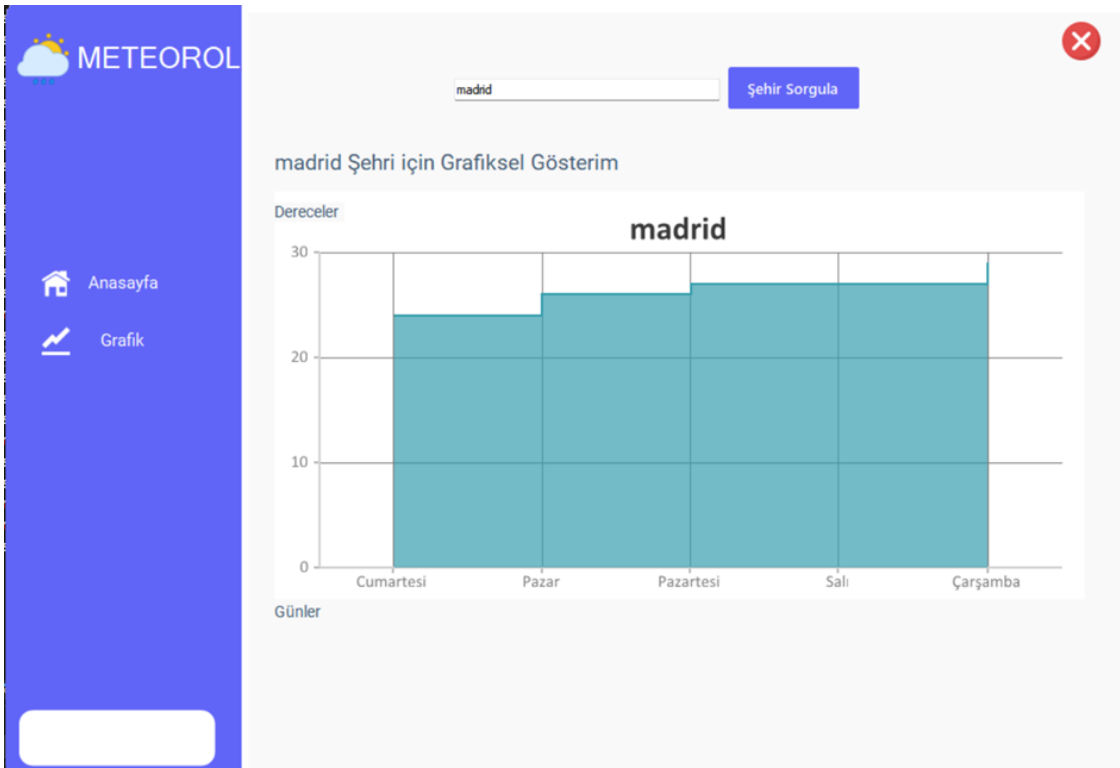
Uygulamada kullanılan interaktif grafikler ve animasyonlar, kullanıcıların hava durumu verilerini daha iyi anlamalarına olanak tanır. Kullanılan grafik metodu haftalık sıcaklık verilerini daha anlaşılır kılmıştır.

Görselleştirmede kullanılan bazı algoritmalar şu şekildedir:

**setIcon Metodu:** Bu metod, hava durumu simgelerini ayarlamak için kullanılır. Gelen iconID parametresi, ikonun kimliğini belirtir. Metod, bu kimliği kullanarak gece veya gündüz simgesine karar verir ve sonrasında OpenWeatherMap servisinin URL'inden bu ikonu çeker. Örneğin, gelen iconID değeri "01d" ise, "01n" şeklinde gece ikonu seçilir. Bu işlem, kullanıcının hava durumu ikonlarını doğru zamanda doğru şekilde görebilmesi için önemlidir. İkonların görsel temsilini sağlayan bu

metod, kullanıcı deneyimini zenginleştirir ve uygulamanın görsel arayüzüne dinamik bir katkı sağlar.

**UnsplashApi Metodu:**Bu metod, kullanıcının bulunduğu şehire özel arka plan resimlerini dinamik olarak ayarlamak için Unsplash API'sini kullanır. Kullanıcı tarafından girilen şehir ismi, API sorgusunda kullanılır ve çeşitli arka plan resimleri bu API üzerinden çekilir. Bu özellik, uygulamanın estetik değerini artırır ve kullanıcıya görsel olarak çekici bir deneyim sunar. Uygulama içinde dinamik arka plan değişikliği sağlayarak kullanıcıların ilgisini çekebilir ve uygulamanın kullanım süresini artırabilir.



Şekil 4.4: Haftalık hava durumu tahmini

Şekil 4.4: Bu ekran, kullanıcıların seçtikleri şehir için 1 haftalık hava durumu tahminini grafiksel olarak gösterir. Görselleştirme, verilerin kolay anlaşılabilir ve erişilebilir olmasını sağlar.

Kullanıcıların seçtikleri şehir için haftalık hava durumu tahminleri grafiksel olarak bu ekran aracılığıyla sunulmaktadır. Görselleştirme, kullanıcıların verileri kolayca anlamasını ve karar vermesini sağlar. Kullanıcılar için tahmin sonuçlarının sunumu, etkileşimli görselleştirmeler aracılığıyla yapılmaktadır.

## Bölüm 5

# Uygulama Mimarisi ve Yönetimi

Bu bölümde, HavaDurumu uygulamasının mimari yapısı ve yönetim süreçleri detaylandırılmaktadır. Uygulama, modüler bir yapıda tasarlanmıştır ve her bir modül, belirli görevleri yerine getirerek toplam işlevselliği desteklemektedir. Aşağıda, uygulamanın önemli bileşenleri ve bunların rolleri üzerine bir inceleme sunulmaktadır.

### **Çözüm Yapısı ve Proje Konfigürasyonları (HavaDurumu.sln)**

Uygulamanın çözüm dosyası, Visual Studio ortamında çeşitli projelerin bir arada yönetilmesini sağlar. HavaDurumu uygulaması, birden fazla projeden oluşan bir yapıya sahiptir, bunlar arasında "HavaDurumu" ve "Meterelog" projeleri yer almaktadır. Çözüm dosyası, her projenin farklı yapılandırmalarını (Debug veya Release gibi) tanımlayarak, geliştirme ve dağıtım süreçlerinde esneklik sağlar. Bu yapılandırmalar, projenin farklı ortamlarda nasıl derleneceğini ve çalıştırılacağını belirler.

### **Dağıtım Yönetimi (HavaDurumu.application)**

Uygulamanın ClickOnce dağıtım manifest dosyası olan HavaDurumu.application, uygulamanın dağıtım ve güncellemeleri ile ilgili bilgileri içerir. Bu dosya, uygulamanın hangi sürümünün dağıtıldığını, kaynak dosyalarını ve güncellemelerin nasıl yönetileceğini tanımlar. Bu sayede kullanıcılar uygulamanın en son sürümünü kullanmaya teşvik edilir ve güncelleme süreçleri otomatik olarak yönetilir.



### **Derleme Bilgileri (AssemblyInfo.cs)**

AssemblyInfo.cs dosyası, uygulamanın derleme sürecinde kullanılan meta verileri içerir. Bu dosya, uygulamanın sürüm numarası, telif hakkı bilgileri, ürün adı gibi bilgileri barındırır ve .NET derlemesi sırasında bu bilgilerin derlemeye dahil edilmesini sağlar. Bu bilgiler, uygulamanın daha iyi tanımlanmasına ve yönetilmesine olanak tanır.

### **Kaynak ve Ayar Yönetimi (Resources.Designer.cs ve Settings.Designer.cs)**

Resources.Designer.cs, uygulamanın görsel ve metinsel kaynaklarını yönetir. Bu otomatik olarak oluşturulan dosya, görsellerin ve diğer kaynakların kod üzerinden kolayca erişilebilir olmasını sağlar. Settings.Designer.cs, uygulama ayarlarını yönetir ve bu ayarların uygulama genelinde tutarlı ve güvenli bir şekilde kullanılmasını sağlar. Bu dosyalar, uygulamanın dinamik ve kullanıcı odaklı olmasını destekler.

### **Kullanıcı Arayüzü Geliştirmeleri (EllipseTools.cs)**

EllipseTools.cs, uygulamanın kullanıcı arayüzünü estetik açıdan geliştiren bir bileşendir. Bu dosya, form elemanlarının köşelerini yuvarlak yaparak modern bir kullanıcı arayüzü sunar. GDI+ kütüphanesini kullanarak yapılan bu işlemler, kullanıcıların arayüzle olan etkileşimlerini zenginleştirir ve uygulamanın görsel çekiciliğini artırır.

*Program.cs:* Uygulamanın ana giriş noktasıdır. Bu dosya, uygulamanın başlangıç ayarlarını yapar ve ana formu yükler. Burası, tüm uygulama için temel yapılandırmaların ve başlangıç işlemlerinin yapıldığı yerdir.

*Anasayfa.cs:* Kullanıcı arayüzünün merkezidir. Tüm kullanıcı etkileşimleri bu dosya üzerinden yönetilir ve uygulamanın ana işlevleri burada tanımlanır. Kullanıcı arayüz elemanları, event handler'lar ve API çağrıları gibi temel işlevler bu dosyada yer alır.

*Gunluk.cs:* Tarih ve zaman işlemleri için özel işlevlerin bulunduğu bir modüldür. Bu dosya, uygulamanın zamanla ilgili işlemlerini yönetir ve zaman verilerini işler.

*Template.cs* ve *Item.cs*: Bu dosyalar, özel kullanıcı kontrol modülleri ve arayüz şablonlarını tanımlar. *Template.cs* genel şablonlar ve stil tanımlamaları için kullanılırken, *Item.cs* özel arayüz bileşenleri ve özellikleri için kullanılır. Her iki dosya da uygulamanın görsel tasarımının ve kullanıcı deneyiminin geliştirilmesinde kritik roller oynar.

Kullanılan bu mimari yapı ve yönetim süreçleri, HavaDurumu uygulamasının etkin bir şekilde geliştirilmesini, yönetilmesini ve dağıtılmasını sağlar. Modüler tasarım ve açıkça tanımlanmış roller, uygulamanın bakımını ve genişletilmesini kolaylaştırırken, entegre araçlar ve otomatik süreçler, geliştirme verimliliğini ve kullanıcı deneyimini iyileştirir. Her modülün belirli bir sorumluluğu olması, kodun bakımını oldukça kolaylar ve yönetimi basitleştirir. Gelecekteki genişletmeler için kolay zemin hazırlar.

# Bölüm 6

## Bulgular

Bu bölüm, projenin teknik altyapısından elde edilen sonuçları ve teorik kullanıcı etkileşimlerini sunmaktadır, özellikle yazılımın nasıl tasarlandığı ve beklenen işlevselliği nasıl yerine getirdiği üzerinde durulmaktadır.

### 6.1 Görselleştirme Ve Arayüz Tasarımı

Uygulama, kullanıcıların hava durumu verilerini kolayca anlamasını ve yorumlamasını sağlamak amacıyla zengin görselleştirme teknikleri kullanmaktadır. İnteraktif grafikler ve şehir bazında detaylı hava durumu görselleri, kullanıcıların verilere hızlı erişimini ve anlık güncellemeleri görmelerini sağlar. Haftalık hava durumu tahminlerinin sunumu, kullanıcıların gelecek günler için plan yapmalarına olanak tanır ve bu bilgiler Şekil 4'te temsil edilir.

### 6.2 Veri İşleme Ve Performans

Uygulama, çeşitli API'lar üzerinden toplanan hava durumu verilerini işleyerek kullanıcıya sunmaktadır. Veri işleme süreci, verilerin temizlenmesi, formatlanması ve normalizasyonunu içerir, böylece kullanıcılara daha doğru ve tutarlı bilgiler sağlanır. Uygulamanın performansı, yüksek verimlilik ve hız gösterecek şekilde

tasarlanmıştır, bu da özellikle yoğun kullanım anlarında bile uygulamanın kararlı ve güvenilir çalışmasını sağlar.

## 6.3 Kullanıcı Deneyimi Ve İnteraktivite

Uygulamanın kullanıcı arayüzü, sezgisel navigasyon ve interaktif öğelerle zenginleştirilmiştir. Kullanıcılar, arayüz üzerinde kolayca gezinebilir ve gerekli bilgilere hızlı bir şekilde erişebilir. Ayrıca, arayüzün kullanıcı dostu tasarımı, geniş bir kullanıcı kitlesinin uygulamayı etkili bir şekilde kullanabilmesini sağlar. Uygulamanın kullanıcı etkileşimi, arayüz tasarımındaki ergonomi ve kullanılabilirlik ilkeleri temel alınarak değerlendirilmiştir.

## 6.4 Teknik Yapı Ve Güvenlik Özellikleri

Uygulama, güvenli veri aktarımı ve kullanıcı bilgilerinin korunması için endüstri standartlarında güvenlik protokolleri ile geliştirilmiştir. Veri şifreleme ve güvenli oturum yönetimi, kullanıcı verilerinin güvenliğini sağlar ve güvenilir bir kullanıcı deneyimi sunar.

# Bölüm 7

## Tartışma

### 7.1 Test Süreçleri Ve Elde Edilen Sonuçlar

Test Metodolojisi: Projede, uygulama üzerinde performans analizleri ve kullanılabilirlik testleri yapılmıştır. Her bir test, uygulamanın kullanılabilirlik kalitesi, veri doğruluğu ve yanıt sürelerini değerlendirmek amacıyla detaylı bir şekilde planlanmıştır.

Elde Edilen Sonuçlar: Testler sonucunda, uygulamada yüksek veri doğruluğu saptanmıştır. Özellikle veri görselleştirme özellikleri ve kullanıcı arayüzü tasarımı gayet başarılı sonuç vermiştir. Yapılan performans iyileştirmeleri sayesinde, uygulamanın yanıt süreleri önemli ölçüde kısaltılmıştır.

### 7.2 Genel Tartışma Ve Gelecek Çalışmalar

Projenin Uygulama Alanları: Uygulama, özellikle acil durum yönetimi, turizm sektörü ve şehir planlamasında büyük potansiyele sahiptir. Hava durumu bilgilerinin doğru ve zamanında sunulması, bu alanlardaki karar verme süreçlerini önemli ölçüde iyileştirebilir.

*Gelecek Çalışmalar için Öneriler:* Gelecekte, uygulamanın yapay zeka ve makine öğrenimi algoritmalarını kullanarak tahmin yeteneklerini geliştirmesi hedeflenmektedir. Ayrıca, daha geniş veri setleri üzerinde çalışarak, modelin doğruluğunu artırma ve yeni kullanıcı özellikleri eklemek de planlar arasındadır.

Örneđin, ekstrem hava olayları sırasında daha detaylı bilgilendirme ve önerilerin sunulması planlanmaktadır.

*Kullanıcı Deneyimi Arařtırmaları:* Uygulamanın farklı kullanıcı grupları üzerindeki etkilerini ölçmek için kapsamlı kullanıcı deneyimi çalışmaları yapılabilir. Bu çalışmalar, uygulamanın ergonomi ve kullanılabilirlik yönlerinin geliştirilmesine katkı sağlar.

*İklim Deđişiklikleri ve Uygulama Performansı:* İklim deđişikliklerinin uygulama üzerindeki etkilerini inceleyen çalışmalar yapılabilir. Bu çalışmalar, uygulamanın farklı iklim koşullarında nasıl optimize edilebileceğine dair bilgiler sunabilir.

# Bölüm 8

## Sonuç ve Öneriler

### 8.1 Sonuçlar

Bu proje, hava durumu verilerinin işlenmesi ve görselleştirilmesi konusunda önemli teknik ilerlemeler sağlamıştır. Geliştirilen uygulama, kullanıcı dostu arayüzü ve etkili görselleştirme teknikleri ile dikkat çekmektedir. Uygulama, modern API'lar üzerinden dinamik olarak hava durumu verilerini çekebilmiş ve bu verileri anlaması kolay, interaktif grafiklerle sunmuştur. Kullanıcı arayüzü, özellikle sade ve anlaşılma kolaylığı sağlayacak şekilde optimize edilmiştir. Performans testleri, uygulamanın yüksek trafik altında bile istikrarlı ve hızlı yanıt verdiğini göstermiştir.

### 8.2 Uygulama Ve Öneriler

**Kapsamın Genişletilmesi:** Uygulamanın coğrafi kapsamının genişletilmesi, daha fazla kullanıcıya hizmet verebilmesini sağlayacaktır. Ayrıca, yerel dil destekleri eklenerek farklı bölgelerdeki kullanıcıların uygulamayı daha rahat kullanmalarını hedeflenebilir.

**Kişiselleştirme ve Adaptasyon:** Kullanıcılara hava durumu verilerini kişiselleştirebilme imkanı sunulabilir. Örneğin, kullanıcılar en çok ilgilendikleri şehirleri kaydedebilir ve bu şehirlerin hava durumu bilgilerine ana sayfadan hızlıca ulaşabilir.

*Gelişmiş Güvenlik Özellikleri:* Özellikle kullanıcı verilerinin depolanması ve işlenmesi süreçlerinde gelişmiş güvenlik protokolleri uygulanmalıdır. Çok faktörlü kimlik doğrulama ve veri şifreleme yöntemleri, kullanıcı bilgilerinin korunmasını güçlendirecektir.

*Kullanıcı Geri Bildirim Mekanizmaları:* Uygulamaya, kullanıcıların geri bildirimde bulunabileceği araçlar entegre edilerek, kullanıcı memnuniyeti ve uygulamanın kullanılabilirliği konusunda sürekli iyileştirmeler yapılabilir.



# Kaynaklar

Few, S. (2009). *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press.

Freeman, E., & Robson, E. (2010). *Head First Design Patterns*. O'Reilly Media.

Microsoft. (2022). .NET Documentation. Retrieved from <https://docs.microsoft.com/en-us/dotnet/>

Murray, S. (2013). *Interactive Data Visualization for the Web*. O'Reilly Media.

Norman, D. (2013). *The Design of Everyday Things: Revised and Expanded Edition*. Basic Books.

OpenWeatherMap. (2022). API Documentation for Weather Data. Retrieved from <https://openweathermap.org/api>

Tufte, E. R. (2001). *The Visual Display of Quantitative Information*. Graphics Press.