

MÜŞTERİ GERİ BİLDİRİMİ VE ANKET SİTESİ

Dr. Ömer ASLAN

Merve BAĞIŞLAR 2111505050

Beyza ATAY 2111505013

Dr. Öğretim Üyesi Ömer Aslan, Yazılım Mühendisliği Öğrencileri

Bandırma Onyedli Eylöl Üniversitesi, Balıkesir/Türkiye

Oaslan@bandirma.edu.tr

mervebagislar07@gmail.com

beyzaaty00@gmail.com

ÖZET

Müşteri geri bildirimi ve anket uygulamalarının işletmeler için kritik bir stratejik araç olduğunu vurgulamaktadır. Müşteri memnuniyeti, işletmelerin başarısı için belirleyici bir faktör olduğundan, müşteri geri bildirimlerini toplamak, analiz etmek ve bu verilere dayanarak hizmet kalitesini iyileştirmek hayati bir öneme sahiptir. Gerçek müşteri görüşleriyle desteklenen bu veriler, işletmelerin rekabet avantajı elde etmelerine ve sürdürülebilir bir müşteri sadakati oluşturmalarına yardımcı olabilir. Bu rapor, müşteri geri bildirimi ve anket uygulamalarının hizmet sektöründe ve diğer endüstrilerde nasıl başarılı bir şekilde uygulanabileceği konusunda yönlendirici bir kaynak olarak tasarlanmıştır. Bu raporda anlatılacak olanlar daha önceden yapılmış anket siteleri, bizim yaptığımız (HotelRate) anket sitesi kodlarıyla beraber uygulamalı olarak açıklamak, projedeki geliştirilecek olan bölümler ve önceki çalışmalarla karşılaştırılması olacaktır.

Anahtar Kelimeler- Müşteri Geri Bildirimi, Anket Uygulamaları, Hizmet Kalitesi, Müşteri Memnuniyeti, İşletme Stratejisi, Otel, anket, veri tabanı, MVC, bootstrap, SQL Server, Html, CSS, JavaScript, C#

CUSTOMER FEEDBACK AND SURVEY PLATFORM

Abstract

Customer feedback and survey applications are emphasized as critical strategic tools for businesses. Since customer satisfaction is a decisive factor for the success of enterprises, collecting, analyzing, and improving service quality based on customer feedback is of vital importance. Surveys stand out as a powerful tool for identifying customer expectations, pinpointing service weaknesses, and enabling businesses to develop customer-focused strategies. This report is designed to serve as a guiding resource on how customer feedback and survey applications can be successfully implemented in the service sector and other industries. The content of this report will include previously conducted survey sites, practical explanations using the codes of our own survey site (HotelRate), the sections to be developed in the project, and a comparison with previous works.

Keywords- Customer Feedback, Survey Applications, Service Quality, Customer Satisfaction, Business Strategy, Hotel, Survey, Database, MVC, Bootstrap, SQL Server, Html, CSS, JavaScript, C#.

1. GİRİŞ(INTRODUCTION)

Müşteri geri bildirimi, işletmeler için değerli bir kaynaktır. Müşterilerin ürün ve hizmetleri hakkındaki düşüncelerini ve duygularını anlamanın bir yoludur. Bu bilgiler, işletmelerin ürün ve hizmetlerini geliştirmelerine, müşteri deneyimini iyileştirmelerine ve rekabette öne çıkmalarına yardımcı olabilir. Müşteri geri bildirimini toplamak için çeşitli yöntemler kullanılabilir. En yaygın yöntemlerden biri anketlerdir. Anketler, müşterilerden belirli soruları yanıtlamalarını isteyerek geri bildirim toplar. Anketler, müşteri memnuniyetini ölçmek, ürün ve hizmet deneyimi hakkında bilgi edinmek ve müşteri ihtiyaçlarını ve isteklerini belirlemek için kullanılabilir.

2. LİTERATÜR TARAMASI

Bu bölümde, mevcut bilimsel çalışmaları ve araştırmaları değerlendirerek, müşteri geri bildirimi ve anket uygulamalarının işletmeler için stratejik önemini destekleyen temel konulara odaklanılacaktır.

2.1 İNTERNET ANKETLERİ

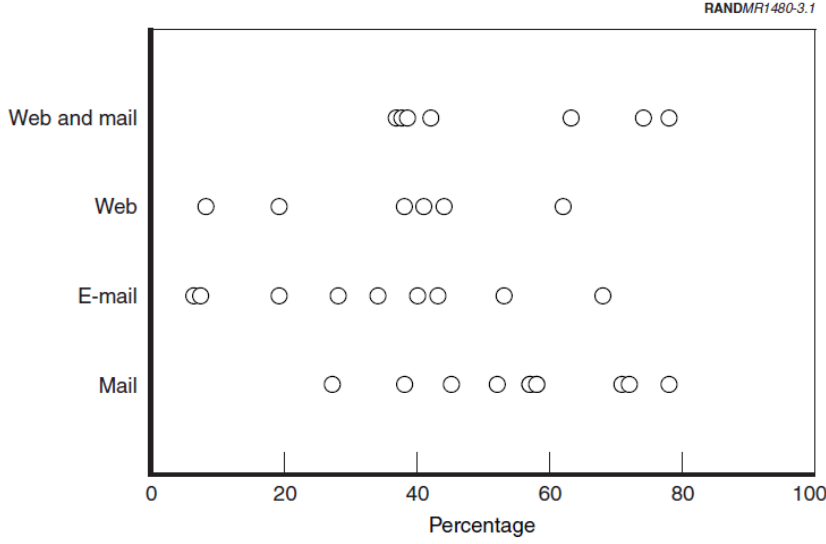
İnternet anketleri, çevrimiçi platformlar aracılığıyla gerçekleştirilen anket türleridir ve katılımcılara çeşitli konularda sorular sorarak geri bildirim toplamak amacıyla kullanılır. İnternet anketleri, geleneksel kağıt-tabanlı anketlere kıyasla daha hızlı, daha ekonomik ve geniş bir katılımcı kitlesine ulaşma potansiyeline sahiptir.

Ancak, internet anketleriyle ilgili bazı zorluklar da vardır. Örneğin, internet erişimine sahip olmayan veya çevrimiçi anketlere katılmaktan kaçınan kişileri dışlayabilir. Ayrıca, örneklem seçiminde önyargıya neden olabilir ve veri güvenilirliğini etkileyebilir. Bu nedenle, araştırmacılar ve anket tasarımcıları, bu potansiyel sorunları göz önünde bulundurmalı ve çözümler üretmelidir.

2.2 İNTERNET ANKETLERİNİN YANIT ORANLARI

İnternet anketlerinin yanıt oranları, anket moduna göre Şekil 1'de grafiksel olarak özetlenmiştir (daha kesin sayılar Ek B'de bulunabilir). Genel olarak, Şekil 1, posta yanıt modu kullanan anketlerin ve posta ile Web yanıt modu kullanan anketlerin, sadece e-posta veya Web yanıt modu kullananlara göre genellikle daha yüksek yanıt oranlarına sahip olduğunu önermektedir.

Web anketleri için yanıt oranları %7 ile %44 arasında değişirken, e-posta anketleri için bu oran %6 ile %68 arasında değişmektedir. Literatürdeki bazı çalışmalar, katılımcılara ya posta ya da İnternet üzerinden yanıt verme seçeneği sunmuştur. İncelediğimiz yedi çalışmanın beşinde katılımcılar genellikle posta yoluyla yanıt vermeyi İnternet üzerinden yanıt vermeye tercih ettiler ve iki çalışma tam tersini buldu. Her şeyden önce, bireysel çalışmanın bağlamının önemli olduğu görülmektedir. Örneğin, en yüksek oranda İnternet yanıtlayıcısı olan çalışmanın katılımcıları, İnternet tartışma gruplarında reklam aracılığıyla toplandı (Schleyer and Forrest, 2000). Literatürdeki birçok çalışma, e-posta anketlerinin posta anketlerine göre daha düşük veya daha yüksek yanıt oranlarına sahip olup olmadığını belirlemek için deneyler yapmayı içeriyor. Bu tür çalışmalarda, aynı nüfusun farklı bölümlerine (veya çalışma kollarına) aynı anket formları gönderildi. Çalışma kolları arasındaki tek fark, katılımcının e-posta yoluyla mı yoksa posta yoluyla mı yanıt vermesi gerektiği idi. Çoğu çalışmada, posta yanıt oranı %21'e kadar daha yüksek bulundu. Yalnızca bir çalışma, daha düşük bir posta yanıt oranına yol açtı. Ancak, bu çalışma birçok açıdan olağandışıydı ve e-postanın yeniliğinin sonuçları etkilemiş olabileceği bir zamanda yapıldı (Parker, 1992).



Şekil 1

2.3 DAHA ÖNCE YAPILAN ÇALIŞMALAR

Müşteri geri bildirimi ve anket alanında gerçekleştirilmiş önceki projeler, bu bölümde detaylı bir şekilde incelenecektir. Bu projeler, müşteri memnuniyeti ölçümü, hizmet kalitesi değerlendirmesi, anket tasarımı, veri analizi, ve sonuçların işletme stratejilerine entegrasyonu gibi çeşitli alanlarda gerçekleşmiş olabilir.

2.3.1 ANKETİX

Bu site kullanıcılara belirli konularda anket soruları sorarak bir otel hakkında geri bildirim almayı hedeflemiştir. Basit bir kullanıcı ara yüzü kullanılmıştır. HTML, CSS projesidir.

Daha detaylı bilgiye aşağıdaki site linkinden ulaşabilirsiniz.

<http://anket.anketix.com/fs.aspx?surveyid=2c549c4879847109bec10b4037e7758>

1. Rezervasyon

	Çok Kötü	Kötü	Orta	İyi	Çok İyi	Fikrim Yok
Yetkiliye ulaşım kolaylığı	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
İlgi ve nezaket	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Otel'e Ulaşım

	Çok Kötü	Kötü	Orta	İyi	Çok İyi	Fikrim Yok
Otel Servisi Yetkilisi Nezaketi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Otel Servisi Bekleme Süresi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Otel Servisi Rahatlığı	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. PERSONEL

Lütfen, Aşağıdaki soruları personelle baz alarak cevaplayınız.

	Çok Kötü	Kötü	Orta	İyi	Çok İyi	Fikrim Yok
Oda Servisi İlgi ve Nezaket	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Oda Servisi Temizlik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Oda Servisi Kılık ve Kiyafet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Temizlikçi İlgi ve Nezaket	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Temizlikçi Temizlik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Temizlikçi Kılık ve Kiyafet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rezervasyon İlgi ve Nezaket	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rezervasyon Temizlik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rezervasyon Kılık ve Kiyafet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aşçı ve Garsonlar İlgi ve Nezaket	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aşçı ve Garsonlar Temizlik	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aşçı ve Garsonlar Kılık ve Kiyafet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. ODA İle İlgili sorular

	Çok Kötü	Kötü	Orta	İyi	Çok İyi	Fikrim Yok
Oda Temizliği	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Resim 1: Anketix arayüzü


Anketimize katıldığınız için teşekkür ederiz.

2.3.2 ONTUR HOTEL MEMNUNİYET ANKETİ

Bu site, otelde konaklayan müşterilerin geri bildirim almak amacıyla hazırlanmıştır. Sorular 7 konuya ayrılarak otel değerlendirilmesi yapılmıştır. Müşteriye ait bilgilerin alınması için bir bölüm ayrılmıştır.

ÇEŞME | ANKARA | İZMİR | İSKENDERUN

Ontur Butik Hotel / Ankara +90 (312) 427 80 00 > EN



OTELİMİZ | KONAKLAMA | RESTAURANT | GALERİ | İLETİŞİM

MEMNUNİYET ANKETİ

GENEL BİLGİLER

Ad

Soyad

Telefon

E-Posta

Adres

Oda Numarası

Otelde Kalış Süresi(Gün)

Giriş Tarihi

Çıkış Tarihi

ÖN BÜRO RESEPSİYON

KVKK İlişkin Başvuru Formu KVKK İlişkin Aydınlatma Metni

Resim 2: ONTUR otel anket arayüzü

ÖN BÜRO RESEPSİYON

	Çok İyi	İyi	Orta
Giriş Çıkış İşlemlerinin Çabukluğu	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
BellBoy ve Bagaj Hizmetleri	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tesis Hakkında Bilgilendirme	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	-	-	-

BAHÇE

Genel Düzenlenme / Peyzaj	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bakım	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bitki Çeşitliliği	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Personelin İlgi ve Nezaketi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Yorumlarınız / Tavsiyeleriniz					

BARLAR - SERVİS

Orta Bar Servis Kalitesi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Havuz Bar Servis Kalitesi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Restoran Bar Servis Kalitesi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Disco Bar Servis Kalitesi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Personelin İlgi ve Nezaketi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Yorumlarınız / Tavsiyeleriniz					

ÖNERİLERİNİZ

	Evet	Hayır			
Otelimizde Daha Önce Bulunuz Mu?	<input type="radio"/>	<input type="radio"/>			
Yanda Belirtilen Elemanlarınızı Sunduğu Hizmet Dolayısıyla Övgüye Değer Buluyorum					
Genel Düşünce ve Önerileriniz					
	Acentam Tavsiye Etti	Daha Önceki Gelişimde Memnun Kalmıştım	Gazete, Dergi vb. İlanlardan	Arkadaşım Tavsiye Etti	İnternette Buldum
Bizi Nereden Duydunuz?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
	Evet	Belki	Hayır		
Tekrar Gelir Misiniz?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>		

Güvenlik Kodu

Güvenlik Kodu

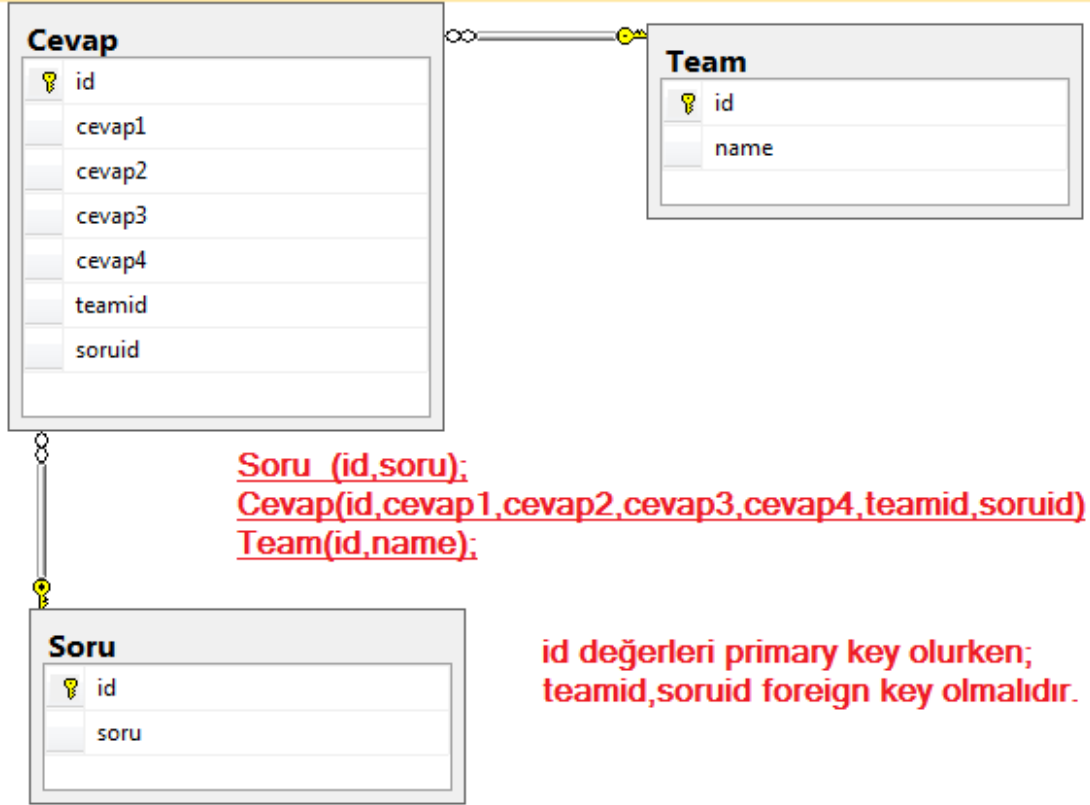
d b t e s

ANKETİ TAMAMLA

Resim 3: Ontur otel anket soruları

2.3.3 En İyi Futbolcu Anket Sistemi

4 büyük futbol takımın 4 tane oyuncusu veriliyor. Seçtiğiniz takımın en iyi oyuncusunun seçilmesi isteniyor .O yüzden her takım için ayrı bir oylama yapılıyor. Soru ve cevap tablosunun yanında bir de takım isimlerini tutan ayrı bir tablo vardır. Tablo diyagramı aşağıdaki gibidir : Veri tabanının adı **anket**'tir



Şekil 2: Sistemin Veritabanı

Tablolar oluşturulduktan sonra **Web.Config** dosyasına connection stringi yerleştirilmeli:

```
<connectionStrings>
<add_name="anket"_connectionString="Data
Source=.\SQLEXPRESS;AttachDbFilename=|DataDirectory|\anket.mdf;Integrated_Security=
True;UseRInstance=True"
providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Yeni bir web form seçilip ismi Anket.aspx yapılır. Daha sonra oluşturduğumuz kullanıcı kontrolünü bu sayfaya sürüklenir. Sürüklendiğinde bu kontrolün sayfadaki id değeri usercontrol1 olur. Daha sonra sayfaya bir buton eklenir. Text değerini Gönder yapılır.

Anket.aspx.cs Anket.aspx X

Takımını Seç : Galatasaray ▼

En sevdiğin oyuncu: Unbound ▼

Gönder

Şekil 3: Web form

Örnek kullanımlar şu şekildedir:

Example 1:
Takımını Seç : Galatasaray ▼
En sevdiğin oyuncu: Arda ▼
Gönder
Galatasarayın en sevilen oyuncusu:
Baros:%8
Arda:%23
Servet:%38
Culio:%31

Example 2:
Takımını Seç : Fenerbahçe ▼
En sevdiğin oyuncu: Alex ▼
Gönder
Fenerbahçenin en sevilen oyuncusu:
Alex:%50
Semih:%17
Dia:%17
Gökhan:%17

Example 3:
Takımını Seç : Beşiktaş ▼
En sevdiğin oyuncu: Quaresma ▼
Gönder
Beşiktaşın en sevilen oyuncusu:
Quaresma:%92
Guti:%0
Simao:%8
Almeida:%0

Şekil 4: Örnek kullanımları

3. KULLANILAN TEKNİKLER

Anket teknikleri, anket sitelerinin kullanıcılarına daha iyi bir deneyim sunmalarını, daha doğru veri toplamalarını ve anket sonuçlarını etkili bir şekilde yönetmelerini sağlar.

- **Kullanıcı Arayüzü (UI) ve Deneyimi (UX) Tasarımı:** Anket siteleri, kullanıcıların kolayca anketleri doldurmasını sağlamak için etkili bir UI/UX tasarımına sahip olmalıdır. Kullanıcı dostu ve anlaşılır bir arayüz, kullanıcıların anketlere daha istekli katılmasını sağlar.
- **Dinamik Formlar ve Sorular:** Anketlerde kullanılan dinamik form teknikleri, katılımcılara verdikleri yanıtlara bağlı olarak soruları otomatik olarak uyarlamak için kullanılır. Bu, anketin kişiselleştirilmesini ve daha etkili bir veri toplama sürecini sağlar.
- **Mobil Uyumlu Tasarım:** Mobil cihazlar üzerinden anketlere erişim sağlamak, kullanıcıların herhangi bir platformda rahatlıkla anketleri doldurmalarını sağlar. Mobil uyumlu tasarım, farklı ekran boyutlarına ve cihaz türlerine uygun olarak optimize edilmelidir.
- **Çoklu Dil Desteği:** Küresel katılımı artırmak adına anket siteleri, kullanıcıların tercih ettikleri dilde anketleri doldurmalarına olanak tanıyan çoklu dil desteği sunmalıdır.
- **Güvenlik Önlemleri:** Anket siteleri, kullanıcı verilerinin güvenliğini sağlamak için çeşitli güvenlik önlemleri almalıdır. Bu, SSL şifreleme, güçlü parola politikaları ve kullanıcı girişi doğrulama gibi unsurları içerebilir.
- **Doğrulama ve Sahtecilik Önleme:** Anket siteleri, katılımcıların kimliklerini doğrulamak ve sahte yanıtları engellemek adına doğrulama yöntemleri kullanabilir. Bu, anket sonuçlarının güvenilir ve temsilci olmasını sağlar.
- **İlerleme Göstergeleri ve Geribildirim:** Kullanıcılara anketin tamamlanma durumu hakkında bilgi veren ilerleme çubukları veya aşama göstergeleri, kullanıcıların motivasyonunu artırabilir. Ayrıca, katılımcılara yanıtlarının alındığına dair anlık geribildirim sağlamak da önemlidir.
- **Veri Analizi ve Raporlama Araçları:** Anket siteleri, toplanan verileri etkili bir şekilde analiz etmek ve raporlamak için çeşitli araçlar sunmalıdır. Bu, araştırmacıların anket sonuçlarını anlamalarına ve stratejik kararlar almalarına yardımcı olur.
- **Anketlerin kişiselleştirilmesi:** Anketleri müşterilere göre kişiselleştirmek, katılımı artırmaya yardımcı olabilir. Anketler, müşterilere yönelik bilgiler içerebilir veya müşterilere özel sorular içerebilir.
- **Anketlerin ödüllendirilmesi:** Anketlere katılan müşterilere ödül vermek, katılımı artırmaya yardımcı olabilir. Anketlere katılan müşterilere indirim, kupon veya hediye gibi ödüller verilebilir.
- **Anketlerin otomatikleştirilmesi:** Anketleri otomatikleştirmek, işletmelerin müşteri geri bildirimini toplamak ve analiz etmek için harcadığı zaman ve çabayı azaltmaya yardımcı olabilir. Anketler, e-posta, web sitesi veya mobil uygulama aracılığıyla otomatik olarak gönderilebilir ve yanıtlar otomatik olarak analiz edilebilir.

3.1 KULLANILAN YAZILIM DİLLERİ

Anket uygulamalarında kullanılan yazılım dilleri çeşitlilik gösterebilir. Ancak, bazı yaygın olarak kullanılan yazılım dilleri şunlardır:

- C++: Anket uygulamalarında kullanılan popüler bir programlama dilidir.
- Ruby / Ruby on Rails: Ruby, basitliği ile öne çıkan bir programlama dilidir ve web uygulamalarının oluşturulmasını kolaylaştıran Ruby on Rails çerçevesiyle birlikte kullanılır.
- Java: Java, anket uygulamalarında yaygın olarak kullanılan bir programlama dilidir.
- Kotlin: Kotlin, Android uygulama geliştirme sürecinde kullanılan bir programlama dilidir.
- C#: C#, anket uygulamalarında kullanılan bir başka popüler programlama dilidir.

- HTML ve CSS: Web tabanlı anket uygulamalarında HTML (Hypertext Markup Language) ve CSS (Cascading Style Sheets) dilleri kullanılır.
- Bu diller, anket uygulamalarında kullanılan bazı yaygın yazılım dilleridir. Ancak, her anket uygulaması farklı gereksinimlere sahip olabilir ve farklı diller kullanabilir.

3.2 ANKET SİSTEMLERİNDEKİ KULLANILAN MODEL KODLARININ ALGORİTMALARI

Genellikle anket sistemlerinde kullanılan model kodları, bir dizi algoritma içerir. Bu algoritmalar, kullanıcıların anket oluşturmasını, yanıtlarını yönetmesini, analiz etmesini ve sonuçları raporlamasını sağlar.

Anket Oluşturma Algoritması:

- Anket oluşturma algoritması, kullanıcının soruları eklemesini, düzenlemesini ve anket akışını belirlemesini sağlar.

Örnek model kod:

```
class Survey:
    def __init__(self, title):
        self.title = title
        self.questions = []
    def add_question(self, question):
        self.questions.append(question)
class Question:
    def __init__(self, text, question_type):
        self.text = text
        self.type = question_type
```

Anket Yanıtlarını Yönetme Algoritması:

- Bu algoritma, kullanıcının anket yanıtlarını depolamasını, düzenlemesini ve analiz etmesini sağlar.

Örnek model kod:

```
class Response:
    def __init__(self, user, answers):
        self.user = user
        self.answers = answers
class ResponseManager:
    def __init__(self):
        self.responses = []
    def add_response(self, response):
        self.responses.append(response)
    def analyze_responses(self):
        pass
```

Kullanıcı Yönetimi Algoritması:

- Bu algoritma, kullanıcıların kaydını oluşturmasını, oturum açmasını ve anketlere yanıt vermesini sağlar.

Örnek model kod:

```
class User:
    def __init__(self, username, password):
        self.username = username
        self.password = password

class UserManager:
    def __init__(self):
        self.users = []

    def add_user(self, user):
        self.users.append(user)

    def authenticate_user(self, username, password):
        # Kullanıcı doğrulama işlemleri
        Pass
```

Analiz ve Raporlama Algoritmaları:

- **Algoritma:** Bu algoritmalar, anket yanıtlarını analiz eder ve sonuçları anlamlı raporlar halinde sunar.

Örnek Model Kod:

```
def analyze_responses(responses):
    # Yanıtları analiz etme işlemleri
    pass

def generate_report(analysis_results):
    # Rapor oluşturma işlemleri
    Pass
```

Bu örnekler, anket sistemlerinde kullanılan temel algoritmaları ve bu algoritmaların basit model kodlarını göstermektedir. Gerçek anket sistemleri daha karmaşık olabilir ve işletmenin ihtiyaçlarına ve kullanılan teknolojiye bağlı olarak farklı algoritmalar içerebilir.

4.HOTELRATE PROJEMİZ

4.1 PROJENİN TANITILMASI

Günümüzde turizm sektörü hızla gelişmekte ve birçok kişi farklı destinasyonlarda konaklama ihtiyacını karşılamak için otelleri tercih etmektedir. Bu bağlamda, projemizde kullanıcıların çeşitli otelleri inceleyip, değerlendirmelerde bulunabilecekleri bir platform geliştirmeyi amaçlıyoruz. Projemizde toplam 12 farklı otel bulunmakta olup, kullanıcılar diledikleri oteli seçerek 7 konuya göre anket sorularını

doldurabileceklerdir. Aynı zamanda bu 7 konu hakkında önceki oylama sonuçları hakkında istatistik bilgisine sahip olabilecekler.

4.2 PROJENİN AMACI

Projenin temel amacı, kullanıcılara geniş bir otel yelpazesi sunarak, onların otel deneyimlerini paylaşmalarını ve bu bilgileri diğer potansiyel konaklama yapmak isteyen kişilere sağlamaktır. Ayrıca, dinamik bir yapıya sahip olan projemizde, otel veya sorular veri tabanına kolayca ekleme yapılabilmesi, sistemin sürekli güncel ve geniş bir içeriğe sahip olmasını sağlayacaktır. Projemiz, kullanıcıların doldurduğu anket verilerini kullanarak istatistiksel analizler sunacaktır. Bu analizler, otellerin genel performansını ve kullanıcı memnuniyetini değerlendirecek, aynı zamanda kullanıcılar için önemli bilgiler sunacaktır.

4.3 KULLANILAN TEKNOLOJİLER

Proje geliştirme sürecinde çeşitli teknolojik araçlar kullanılarak, platformun başarılı bir şekilde oluşturulması hedefledik. Aşağıda, projenin temel teknolojik bileşenleri sıralanmıştır:

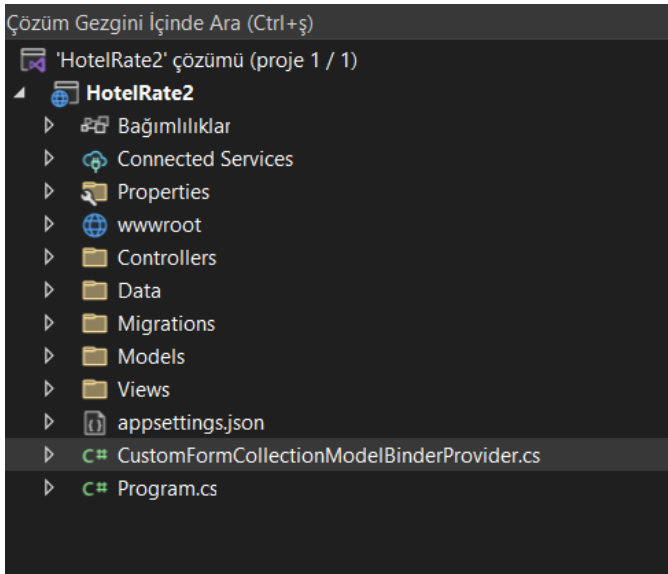
- Frontend: Projenin kullanıcı ara yüzü için Bootstrap, HTML, CSS ve JavaScript kullanılmıştır. Kullanıcıların web sitesi üzerinden etkileşimde bulunacakları arayüz bu teknolojilerle tasarlanmıştır.
- Backend: Sunucu tarafında, kullanıcı verilerini işleyecek ve depolayacak bir backend altyapısı oluşturulmuştur. ASP .NET CORE MVC mimarisine uygun şekilde tasarlanmıştır.
- Veri tabanı: Kullanıcı verileri, sorular, cevaplar ve otel bilgileri için uygun bir veri tabanı sistemi MSSQL ile kurulmuştur.
- İstatistik: SQL sorguları kullanılmıştır.
- Güvenlik: Projenin güvenliği için SSL sertifikaları kullanılacak ve kullanıcı bilgileri güvenli bir şekilde depolanır.

4.4 PROJENİN UYGULAMALI AÇIKLAMASI

Bu bölümde proje kodlarıyla birlikte açıklanacaktır. Proje MVC mimarisine uygun bir şekilde oluşturulmuştur.

4.4.1 Proje dosyaları

Projemizin dosya yapısı MVC formatına uygun bir şekilde oluşturuldu.



Resim 4: Proje dosyaları

4.4.2 Veri tabanı

Bu projede connection strings kullanarak veri tabanı bağlantısını gerçekleştirdik.

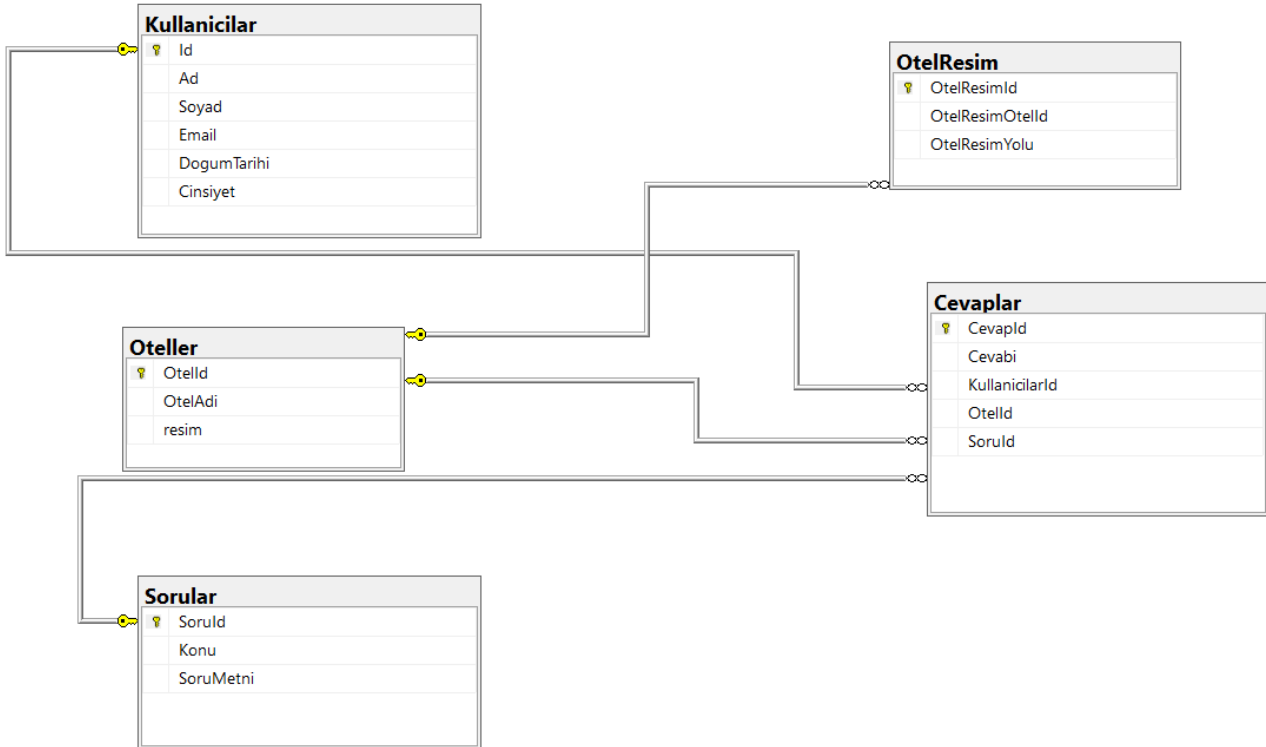
```
"Logging": {  
  "LogLevel": {  
    "Default": "Information",  
    "Microsoft.AspNetCore": "Warning"  
  }  
}  
  
"ConnectionStrings": {  
  "DefaultConnection": "Server=DESKTOP-RLSD01H\\SQLEXPRESS;Database=HotelDB;Trusted_Connection=True;TrustServerCertificate=True"
```

Resim 5: Veritabanı bağlantısı

Veri tabanı tablosu, kullanıcılar, oteller, sorular, cevaplar, OtelResim tablolarını içermektedir.

- Kullanıcılar tablosu: PrimaryKey(Id), Ad, Soyad, Email,Dogum Tarihi, Cinsiyet verilerini tutar.
- Oteller tablosu: PrimaryKey(OtelId), OtelAdı, resim verilerini tutar.
- Sorular tablosu: PrimaryKey(SoruId), Konu ve Soru metni verilerini tutar.
- OtelResim tablosu: PrimaryKey(OtelResimId), OtelResimYolu verilerini ve Otel Idyi foreign key olarak tutar.
- Cevaplar tablosu: Primarykey(CevapId),Cevabi ve foreign key olarak kullanicilarId,OtelId, SoruId verilerini tutar.

Aşağıda veri tabanı tabloları ve ilişkileri gösterilmiştir.



Şekil 5: HotelRate veritabanı sistemi

4.4.3 Program.cs Dosyası

MVC (Model-View-Controller) desenine sahip .NET uygulamamızda, Program.cs dosyası uygulamanın ilk olarak çalıştırıldığı yerdir. Bu dosyada uygulamanın temel yapılandırması ve ana giriş noktası bulunuyor.

WebApplication.CreateBuilder(args) Metodu:

- **WebApplication.CreateBuilder(args)** metodu, bir **WebApplicationBuilder** örneği oluşturarak ASP.NET Core uygulamasının yapılandırılmasına başlar.

builder.Services.AddControllersWithViews(...) Metodu:

- **AddControllersWithViews** metodu, MVC (Model-View-Controller) yapılandırmasını ekler.
- Bu metot içinde, özel bir model bağlama sağlayıcısı (**CustomFormCollectionModelBinderProvider**) eklenmiştir.

builder.Services.AddDbContext<HotelDbContext>(...) Metodu:

- **AddDbContext** metodu, uygulamanın veri tabanına erişim sağlamak için Entity Framework Core'u yapılandırır.
- Bu örnekte, **HotelDbContext** tipinden bir veri tabanı bağlaması eklenir. Bağlantı dizesi **DefaultConnection** adlı bir bağlantı dizesiyle alınır.

app.UseStaticFiles(...) Metodu:

- **UseStaticFiles** metodu, uygulamanın statik dosyaları (örneğin, CSS, JavaScript, resimler) servis etmesini sağlar.
- Bu örnekte, **wwwroot** klasöründeki dosyaların, uygulama kök dizininde **wwwroot** yolunda sunulmasını sağlar.

app.MapControllerRoute(...) Metodu:

- **MapControllerRoute** metodu, MVC uygulamasında controller'ları ve action'ları yönlendirmek için kullanılır.
- Bu örnekte, varsayılan olarak "Giris" controller'ının "Index" action'ına yönlendirme yapılır.

app.Run() Metodu:

- **Run** metodu, uygulamanın çalıştırılmasını sağlar. Uygulama başladığında bu metot çağrılır.

```
using HotelRate2.Models;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);
builder.Services.AddSession();
builder.Services.AddControllersWithViews(
    options =>
    {
        options.ModelBinderProviders.Insert(0, new CustomFormCollectionModelBinderProvider());
    });

builder.Services.AddDbContext<HotelDbContext>(option => option.UseSqlServer(
    builder.Configuration.GetConnectionString("DefaultConnection")

));

var app = builder.Build();

app.UseSession();

app.UseStaticFiles(new StaticFileOptions()
{
    RequestPath = "/wwwroot"
});

app.MapControllerRoute("main", "{controller=Giris}/{action=Index}/{id?}");

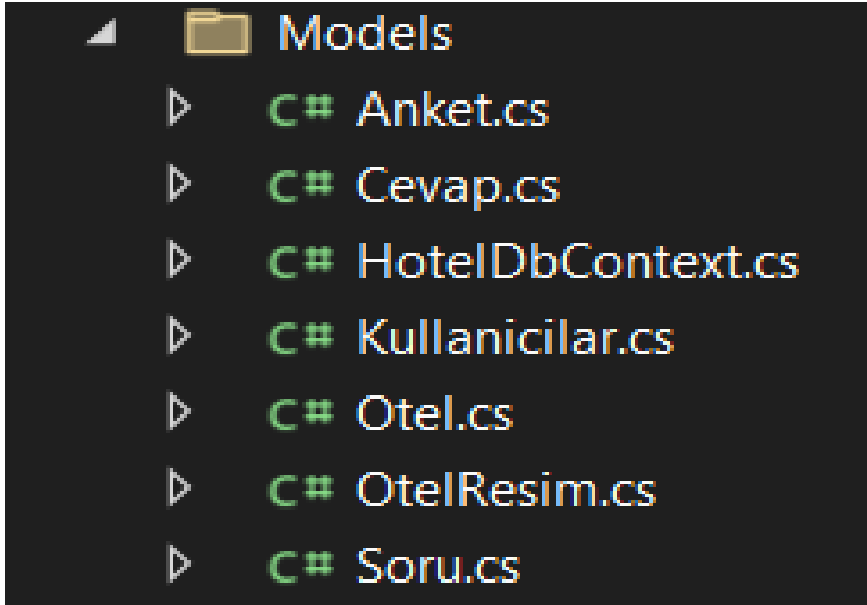
app.Run();
```

Resim 6: Program.cs sınıfı

4.4.5 MODELLER

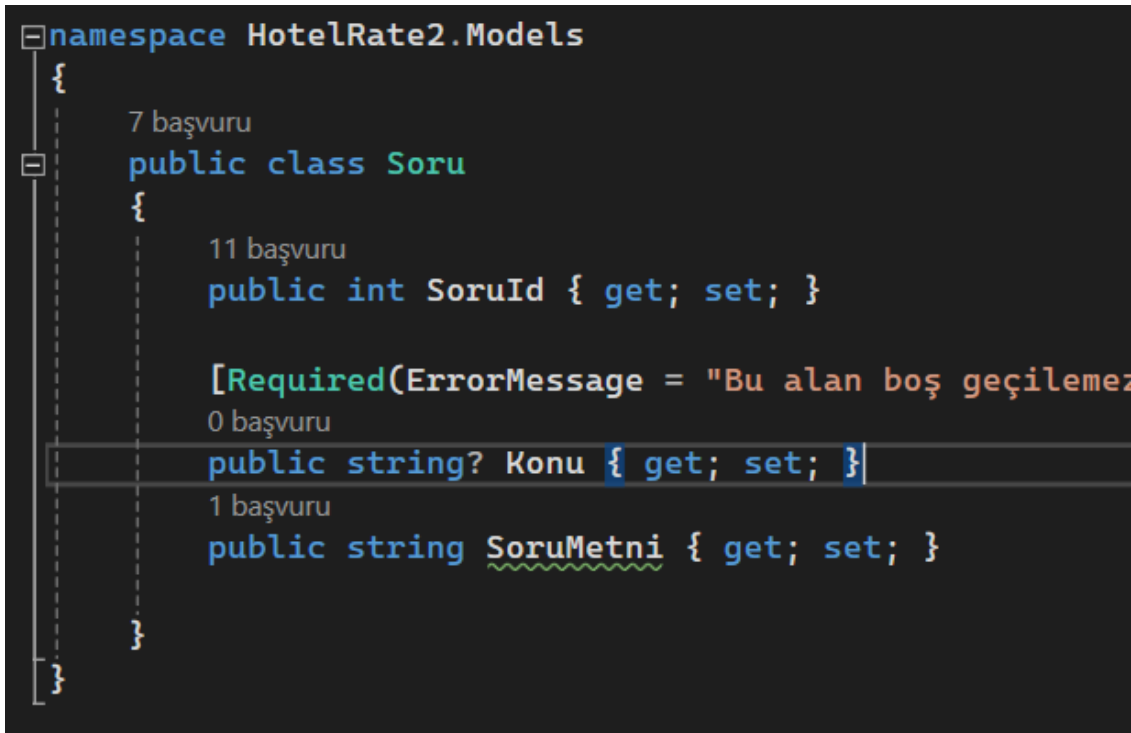
Modeller projemizde veri tabanındaki tabloları temsil ediyor.

Aşağıda model klasörünün altındaki sınıflar gösterilmiştir.



Resim 7: Models sınıfları

Sorular Modeli:



Resim 8: Sorular Modeli

Cevap Modeli:

```
5 başvuru
public class Cevap
{
    0 başvuru
    public int CevapId { get; set; }
    1 başvuru
    public string Cevabi { get; set; }

    0 başvuru
    public int KullaniciId { get; set; }

    0 başvuru
    public Kullanici kullanici { get; set; }
    0 başvuru
    public int OtelId { get; set; }
    0 başvuru
    public Otel Otel { get; set; }

    1 başvuru
    public int SoruId { get; set; }
    0 başvuru
    public Soru Soru { get; set; }

    1 başvuru
    public Cevap(int soruId, string cevabi)
    {
        Cevabi = cevabi;
        SoruId = soruId;
    }
}
```

Resim 9: Cevap Modeli

Otel Modeli:

```
namespace HotelRate2.Models
{
    12 başvuru
    public class Otel
    {
        2 başvuru
        public int OtelId { get; set; }
        [Required(ErrorMessage = "Bu alan boş geçilemez!")]
        4 başvuru
        public string? OtelAdi { get; set; }

        2 başvuru
        public List<OtelResim> OtelResimleri { get; set; }

        2 başvuru
        public string resim { get; set; }
    }
}
```

Resim 10: Otel Modeli

Kullanıcılar Modeli

```
namespace HotelRate2.Models
{
    0 başvuru
    public enum Cinsiyet
    {
        Kadın,
        Erkek,
        Diğer
    }
    7 başvuru
    public class Kullanicilar
    {
        0 başvuru
        public int Id { get; set; }

        [Required(ErrorMessage = "Bu alan boş geçilemez!")]
        2 başvuru
        public string? Ad { get; set; }
        [Required(ErrorMessage = "Bu alan boş geçilemez!")]
        2 başvuru
        public string? Soyad { get; set; }
        [Required(ErrorMessage = "Bu alan boş geçilemez!")]
        2 başvuru
        public string? Email { get; set; }
        [Required(ErrorMessage = "Bu alan boş geçilemez!")]
        2 başvuru
        public DateTime? DogumTarihi { get; set; }
        [Required(ErrorMessage = "Bu alan boş geçilemez!")]

        3 başvuru
        public int Cinsiyet { get; set; }
    }
}
```

Resim 11: Kullanıcılar Modeli

Otel Resimleri Modeli:

```
3 başvuru
public class OtelResim
{
    0 başvuru
    public int OtelResimId { get; set; }

    0 başvuru
    public Otel Otel { get; set; }

    1 başvuru
    public string OtelResimYolu { get; set; }
}
```

Resim 12: Otel Resimleri Modeli

HotelDbContext Sınıfı:

- Bu kodda , Entity Framework Core kullanılarak bir veri tabanı bağlantısı kurmak için kullanılan bir DbContext (Veri tabanı Bağlamı) sınıfını ekledik. HotelDbContext, **DbContext** sınıfını miras alır. Bu, Entity Framework Core tarafından sağlanan temel sınıftır ve veri tabanı işlemlerini yönetmek için gerekli olan temel özelliklere sahiptir.
- Buradaki constructor yapısı **DbContextOptions<HotelDbContext>** türünden bir parametre alır. DbContext'in nasıl yapılandırılacağını belirler. Bu parametre base class'a iletilir.
- DbSet, veri tabanındaki bir tabloya karşılık gelen bir C# sınıfını temsil eder. Bu sınıflar, veri tabanındaki tabloların veri modelini oluşturur. Bizim projemizdeki Kullanıcılar, Oteller, Sorular, Cevaplar, ve Resimler adlı DbSet özellikleri, bu DbContext ile ilişkilendirilmiş veri tabanındaki farklı tablolara karşılık gelir.
- base.SaveChanges() ifadesi, DbContext'in temel sınıfındaki aynı adlı metodu çağırarak gerçek kaydetme işlemini gerçekleştirir. Bu özel SaveChanges metodunun, olası hatalarını ele almak için bir try-catch bloğu ekledik. Hata durumunda, isteğe bağlı olarak özel bir işlem gerçekleştirilebilir veya hata gösterebilir.

```
namespace HotelRate2.Models;

23 başvuru
public partial class HotelDbContext : DbContext
{
    0 başvuru
    public HotelDbContext(DbContextOptions<HotelDbContext> options)
        : base(options)
    {
    }

    1 başvuru
    public DbSet<Kullanici> Kullanici => Set<Kullanici>();
    3 başvuru
    public DbSet<Otel> Oteller => Set<Otel>();

    1 başvuru
    public DbSet<Soru> Sorular => Set<Soru>();

    1 başvuru
    public DbSet<Cevap> Cevaplar => Set<Cevap>();

    0 başvuru
    public DbSet<OtelResim> Resimler => Set<OtelResim>();

    2 başvuru
    public override int SaveChanges()
    {
        try
        {
            return base.SaveChanges();
        }
        catch (Exception ex)
        {
            throw;
        }
    }
}
```

Resim 13: HotelDbContext Sınıfı

4.4.6 GİRİŞ SAYFASI

Burada kullanıcılardan ad, soyad, email, doğum tarihi ve cinsiyet bilgileri istenir ve bu bilgiler veri tabanına kaydedilir.

Controller:

- Bu controllerı giriş işlemlerini yönetmek için oluşturduk.
- Yapıcıcı metodumuz, **HotelDbContext** türünden bir parametre alır. Bu parametre, veri tabanına erişim sağlamak için kullanılır.
- **Index** metodu, kullanıcıya giriş sayfasını gösterir. HTTP GET isteği geldiğinde çalışır ve "Index" view'ını döndürür.
- Index Metodu (HTTP POST), HTTP POST isteği geldiğinde çalışır. Bu genellikle bir formun gönderildiği durumu temsil eder. Metot, bir Kullanıcılar modeli alır. Bu model, kullanıcının giriş yaparken girdiği bilgileri temsil eder. Kullanıcılar modeli, veri tabanındaki Kullanıcılar tablosuna karşılık gelir. Modelin içindeki bilgiler, kullanıcının giriş yapması için kullanılır.
- try-catch bloğu içinde, Kullanıcılar modeli veri tabanına eklenir (`_context.Kullanıcılar.Add(model)`) ve değişiklikler kaydedilir (`_context.SaveChanges()`). HttpContext.Session ile kullanıcıIdye göre oturum yönetimi sağlanır.
- Eğer bir hata oluşursa, catch bloğu içinde hata işlenir.
- Giriş başarılı olduktan sonra, kullanıcıyı başka bir sayfaya yönlendirmek için RedirectToAction kullanılır. Anasayfa controllerındaki, Index actionuna yönlendirir.

```
using HotelRate2.Models;
using Microsoft.AspNetCore.Http;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using System;

namespace HotelRate2.Controllers
{
    1 başvuru
    public class GirişController : Controller
    {
        private readonly HotelDbContext _context;

        0 başvuru
        public GirişController(HotelDbContext context)
        {
            _context=context;
        }

        0 başvuru
        public IActionResult Index()
        {
            return View();
        }

        [HttpPost]
        0 başvuru
        public IActionResult Index(Kullanıcılar model)
        {
            try
            {
                _context.Kullanıcılar.Add(model);
                _context.SaveChanges();
                HttpContext.Session.SetInt32("KullanıcıId", model.Id);
                TempData["kid"] = model.Id;
            }
            catch (Exception ex)
            {
                ex = ex;
            }
            return RedirectToAction("Index", "Anasayfa");
        }
    }
}
```

Resim 14: HotelRate Controller

View:

- Index sayfasının viewı , bir kullanıcının otel deneyimlerini paylaşmak için bilgilerini girebileceği bir form içeren bir web sayfasını temsil ediyor.

Head Bölümü:

- Meta etiketleri, sayfanın karakter setini ve görüntüleme özelliklerini belirtir.
- Sayfanın başlığı (<title>) "HotelRate" olarak belirlenmiştir.
- Google Fonts ve Vendor CSS dosyaları için bağlantılar bulunmaktadır.
- Favicon, sayfanın sekme başlığındaki ikonu temsil eder.

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@model HotelRate2.Models.Kullanicilar

<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>HotelRate</title>
  <meta content="" name="description">
  <meta content="" name="keywords">
  <!-- Favicons -->
  <link href="wwwroot/img/bannerIcon.png" rel="icon">
  <!-- Google Fonts -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=block" rel="stylesheet">
  <!-- Vendor CSS Files -->
  <link href="wwwroot/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="wwwroot/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="wwwroot/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
  <link href="wwwroot/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="wwwroot/vendor/aos/aos.css" rel="stylesheet">
  <!-- Template Main CSS File -->
  <link href="wwwroot/css/main.css" rel="stylesheet">
</head>
<body>
```

Resim 15: Index View(1)

Header (Başlık):

- Sayfanın üst kısmında bir header bölümü bulunmaktadır.
- "HotelRate" logosu ve sayfa içinde gezinmeyi sağlayan bir menü (**navbar**) bulunmaktadır.

```
<header id="header" class="header d-flex align-items-center fixed-top">
  <div class="container-fluid d-flex align-items-center justify-content-between">
    <a href="index.html" class="logo d-flex align-items-center me-auto me-lg-0">
      
      <h1>HotelRate</h1>
    </a>
    <nav id="navbar" class="navbar px-4">
      <ul>
        <li><a asp-controller="Iletisim" asp-action="Index">İletişim</a></li>
      </ul>
    </nav>
    <!-- navbar -->
  </div>
</header>
```

Resim 16: Index View(2)

Hero Section (Hero Bölümü):

- Sayfanın ana içeriği, büyük bir başlık, açıklama ve bir giriş formunu içeren bir hero bölümü.
- Kullanıcıdan alınacak bilgiler, bir form içinde düzenlendi. Bu bilgilerin yanında hoş geldiniz mesajları ve açıklamalar ekledik.

```
<!-- ===== Hero Section ===== -->
<section id="hero" class="hero d-flex flex-column justify-content-center align-items-center" data-aos="fade" data-aos-delay="1500">
  <!-- Section: Design Block -->
  <div class="background-radial-gradient overflow-hidden">
    <div class="container px-4 py-5 px-md-5 text-center text-lg-start my-5">
      <div class="row gx-lg-5 align-items-center mb-5">
        <div class="col-lg-6 mb-5 mb-lg-0" style="z-index: 10">
          <h1 class="my-5 display-5 fw-bold ls-tight" style="color: ■hsl(218, 81%, 95%)">
            HOTEL RATE <br />
            <span style="color: ■hsl(118, 94%, 40%, 0.603)">Hoş Geldiniz! Otel Deneyimlerinizi Paylaşın!</span>
          </h1>
          <p class="mb-4 opacity-70 fs-18" style="color: ■hsl(218, 81%, 85%)">
            Merhaba sevgili gezgin! Biz, sizin otel deneyiminizin değerli olduğuna inanan bir topluluğuz. Sizlerin deneyimleri, diğer gezginlere ışık tutuyor ve bir sonraki tatil planlarını şekillendiriyor. Şimdi, son konaklamanızın ayrıntılarını paylaşarak hem diğer gezginlere rehberlik edin hem de muhteşem fırsatları yakalama şansını elde edin!
          </p>
        </div>
      </div>
    </div>
  </div>
```

Resim 17: Index View(3)

```
<div class="col-lg-6 mb-5 mb-lg-0 position-relative">
  <div id="radius-shape-1" class="position-absolute rounded-circle shadow-5-strong"></div>
  <div id="radius-shape-2" class="position-absolute shadow-5-strong"></div>
  <div class="card bg-glass">
    <div class="card-body px-4 py-5 px-md-5">
      <form asp-action="Index" asp-controller="Giris" method="post" role="form">
        <div class="col-md-6 mb-4">
          <div class="form-outline">
            <input asp-for="Ad" required type="text" class="form-control" />
            <label asp-for="Ad" class="form-label"></label>
          </div>
        </div>
        <div class="col-md-6 mb-4">
          <div class="form-outline">
            <input asp-for="Soyad" required type="text" class="form-control" />
            <label asp-for="Soyad" class="form-label"></label>
          </div>
        </div>
        <div class="col-md-6 mb-4">
          <input asp-for="Email" type="text" class="form-control" />
          <label asp-for="Email" class="form-label"></label>
        </div>
        <div class="form-outline mb-4">
          <input asp-for="DogumTarihi" type="date" class="form-control text-center" />
          <label asp-for="DogumTarihi" class="form-label">Doğum Tarihi</label>
        </div>
        <div class="form-outline mb-4">
          <h6 class="mb-0 me-4">Cinsiyet:</h6>
          <div class="form-check form-check-inline mb-0 me-4">
            <input asp-for="Cinsiyet" class="form-check-input" type="radio" id="femaleGender" value="0" />
            <label class="form-check-label" for="femaleGender">Kadın</label>
          </div>
          <div class="form-check form-check-inline mb-0 me-4">
            <input asp-for="Cinsiyet" class="form-check-input" type="radio" id="maleGender" value="1" />
            <label class="form-check-label" for="maleGender">Erkek</label>
          </div>
          <div class="form-check form-check-inline mb-0 me-4">
            <input asp-for="Cinsiyet" class="form-check-input" type="radio" id="otherGender" value="2" />
            <label class="form-check-label" for="otherGender">Diğer</label>
          </div>
        </div>
        <input type="submit" id="btn-1" value="Otel Seç ve Ankete Katıl" />
      </form>
    </div>
  </div>
```

Resim 18: Index View(4)

Footer (Altbilgi):

- Sayfanın alt kısmında bir footer bölümü bulunmaktadır.
- Telif hakkı bilgisi ve sitemizin adı bu bölümde yer almaktadır.

```
<!-- ===== Footer ===== -->
<footer id="footer" class="footer">
  <div class="container">
    <div class="copyright">
      &copy; Copyright <strong><span>HotelRate</span></strong>. All Rights Reserved
    </div>
    <div class="credits">
      Designed by <a href="">HotelRate</a>
    </div>
  </div>
</footer>
<!-- End Footer -->
<a href="#" class="scroll-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>
<div id="preloader">
  <div class="line"></div>
</div>
```

Resim 19: Index View(5)

Vendor JS Files (JavaScript Dosyaları):

- Sayfanın en altında, farklı vendor'ların JavaScript dosyaları referans edilmiştir. Bootstrap, Swiper, Lightbox ve AOS gibi kütüphaneler burada yer alır.
- Bu dosyalar, sayfanın görsel efektlerini ve dinamik özelliklerini yönetmek için kullanılır.

```
<!-- Vendor JS Files -->
<script src="wwwroot/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="wwwroot/vendor/swiper/swiper-bundle.min.js"></script>
<script src="wwwroot/vendor/glightbox/js/glightbox.min.js"></script>
<script src="wwwroot/vendor/aos/aos.js"></script>
<script src="wwwroot/vendor/php-email-form/validate.js"></script>
```

Resim 20: Index View(6)

Template Main JS File (Ana JavaScript Dosyası):

- Sayfanın temel JavaScript dosyası, özelleştirilmiş işlevselliği yönetmek ve sayfa davranışlarını kontrol etmek için kullanılır.

```
<script src="wwwroot/js/main.js"></script>
```

Resim 21: Index View(7)

Giriş Sayfasının Görünümü:

Aşağıda örnek bir kullanıcı girişi gösterilmiştir.

HOTEL RATE
Hoş Geldiniz! Otel Deneyimlerinizi Paylaşın!

Merhaba sevgili gezgin! Biz, sizin otel deneyiminizin değerli olduğuna inanan bir topluluğuz. Sizlerin deneyimleri, diğer gezginlere ışık tutuyor ve bir sonraki tatil planlarını şekillendiriyor. Şimdi, son konaklamanızın ayrıntılarını paylaşarak hem diğer gezginlere rehberlik edin hem de muhteşem fırsatları yakalama şansını elde edin!

Ad: Ömer
Soyad: Aslan
Email: oaslan@bandirma.edu.tr
Doğum Tarihi: 12.06.1985
Cinsiyet: ☐ Kadın ☒ Erkek ☐ Diğer
[Otel Seç ve Ankete Katıl](#)

Resim 22: Giriş Sayfası

Veri tabanında kullanıcılar tablosuna eklendi.

DESKTOP-RLSD01H\S...- dbo.Kullanıcılar						
	Id	Ad	Soyad	Email	DogumTarihi	Cinsiyet
	1	sdfsaf	fdf	beyzaaty00@gmail.com	2023-12-22 00...	1
	2	Beyza	Atay	beyzaaty00@gmail.com	2003-05-26 00...	0
	3	Merve	Bağışlar	mervebagislar07@gmail.c...	2002-02-17 00...	0
	4	Beyza	Atay	beyzaaty00@gmail.com	2023-05-31 00...	0
	5	Merve	Bağışlar	mervebagislar07@gmail.c...	2023-12-14 00...	2
	6	Ömer	Aslan	oaslan@bandirma.edu.tr	1985-06-12 00...	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Resim 23: Kullanıcılar Tablosu

4.4.7 ANASAYFA

Kullanıcılar anasayfada dilediği oteli seçerek anketine katılabilirler.

Controller:

- AnasayfaController Veri tabanından otel bilgilerini çeker ve bu bilgileri view'a göndererek kullanıcının görmesini sağlar
- AnasayfaController sınıfının yapıcı metodu aynı Giriş Controller sınıfında olduğu gibi, HotelDbContext tipinden bir parametre alır. Bu parametre, veri tabanına erişim sağlamak için kullanılır.
- Index metodu, web uygulamasının anasayfa olarak kullanılacak görünümü oluşturur. Parametre olarak bir nullable int tipi alır (int? id). `_context.Oteller.ToList()` ifadesi ile veri tabanındaki "Oteller" tablosundaki tüm kayıtları liste olarak alır. Bu liste, "otel" değişkenine atanır.
- Sonrasında, bu otel listesi, "Index" view'ına gönderilir. Bu view, kullanıcıya otellerin listesini göstermek için kullanılır.

```
using HotelRate2.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;

namespace HotelRate2.Controllers
{
    1 başvuru
    public class AnasayfaController : Controller
    {
        private readonly HotelDbContext _context;
        0 başvuru
        public AnasayfaController(HotelDbContext context)
        {
            _context = context;
        }

        0 başvuru
        public IActionResult Index(int? id)
        {
            var otel = _context.Oteller.ToList();
            return View(otel);
        }
    }
}
```

Resim 24: Anasayfa Controller

View:

- @addTagHelper direktifi, Razor sayfasında tag helper'ları kullanabilmek için eklenmiştir. Tag helper'lar, HTML etiketlerini oluşturmak ve işlemek için kullanılan özel etiketlerdir.
- @model List<HotelRate2.Models.Otel>, @model direktifi, sayfanın model tipini belirtir. Bu sayede sayfa, belirtilen model tipinden bir liste alabilir.
- HTML başlık etiketi (<head> içindeki <title>) ve meta etiketleri, sayfanın başlık ve açıklamasını belirtir.
- Favicon ve Google Fonts Bağlantıları bölümünde, Sayfanın favicon'u ve Google Fonts bağlantıları eklenmiştir.
- Vendor CSS ve JS Bağlantıları bölümünde, dış kaynaklı CSS ve JavaScript dosyalarının bağlantıları, sayfanın görünümünü ve davranışını şekillendiren stil ve işlevselliği sağlar.

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@model List<HotelRate2.Models.Otel>
<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>HotelRate</title>
  <meta content="Site açıklaması" name="description">
  <meta content="Anahtar kelimeler girilecek" name="keywords">
  <meta http-equiv="X-UA-Compatible" content="IE=7">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <!-- Favicons -->
  <link href="/wwwroot/img/bannerIcon.png" rel="icon">
  <!-- Google Fonts -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700" rel="stylesheet">
  <!-- Vendor -->
  <link href="/wwwroot/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="/wwwroot/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="/wwwroot/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
  <link href="/wwwroot/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="/wwwroot/vendor/aos/aos.css" rel="stylesheet">
  <!-- Template Main CSS File -->
  <link href="/wwwroot/css/main.css" rel="stylesheet">
</head>
<body>
```

Resim 25: Index View(1)

- Header Bölümü, sayfanın üst kısmındaki navigasyon menüsü ve logo içerir.
- Main Bölümü, sayfanın ana içeriğini oluşturan bölüm. Bu bölümde bir sayfa başlığı ve galeri bölümü bulunmaktadır.

```
<header id="header" class="header d-flex align-items-center fixed-top">
  <div class="container-fluid d-flex align-items-center justify-content-between">
    <a href="index.html" class="logo d-flex align-items-center me-auto me-lg-0">
      
      <h1>HotelRate</h1>
    </a>
    <nav id="navbar" class="navbar px-4">
      <ul>
        <li><a asp-controller="Anasayfa" asp-action="Index">Anasayfa</a></li>
        <li><a asp-controller="Istatistik" asp-action="Index">İstatistik</a></li>
        <li><a asp-controller="iletisim" asp-action="Index">İletişim</a></li>
      </ul>
    </nav><!-- .navbar -->
    <i class="mobile-nav-toggle mobile-nav-show bi bi-list"></i>
    <i class="mobile-nav-toggle mobile-nav-hide d-none bi bi-x"></i>
  </div>
</header><!-- End Header -->
<main id="main" data-aos="fade" data-aos-delay="1500">
  <!-- ===== End Page Header ===== -->
  <div class="page-header d-flex align-items-center">
    <div class="container position-relative">
      <div class="row d-flex justify-content-center">
        <div class="col-lg-6 text-center">
          <h2>Otel Seç ve Oyla !</h2>
          <p>Sizin deneyiminiz başkalarının rehberi olabilir! Şimdi dilediğiniz oteli seçerek anketimize başlayın ve topluluğumuzun</p>
        </div>
      </div>
    </div>
  </div>
```

Resim 26: Index View(2)

- Foreach döngüsünde veri tabanındaki otel tablosunda otel Id'ye göre otel resmi ve oteller çekilmiştir. Bu döngü sayesinde daha dinamik bir sistem kurmuş oluyoruz.

```
<section id="gallery" class="gallery">
  <div class="container-fluid text-center">
    <div class="row gy-4 justify-content-center text-center">

      @foreach (var item in Model)
      {
        <div class="col-xl-3 col-lg-4 col-md-6">
          <strong>@item.OtelAdi</strong>
          <div class="gallery-item h-100">
            @if (item.resim != null)
            {
              
            }
            else
            {
              <span>Resim yok</span>
            }
            <div class="gallery-links d-flex align-items-center justify-content-center text-center">
              <a asp-controller="OtelAnket" asp-action="HotelDetay" asp-route-id="@item.OtelId"><strong>Ankete Katıl</strong></a>
            </div>
          </div>
        </div>
      }
    </div>
  </div>
</section>
```

Resim 27: Index View(3)

- Projemizde kullandığımız ve veri tabanına kaydedilen oteller ve resimleri aşağıdaki gibidir.

OtelId	OtelAdi	resim
1	GrandAsya Hotel	grandasya01.jpg
2	Çırağan Palace Hotel	ciragan3.jpg
3	Four Seasons Hotel	fourseasonsist1.jpg
4	Ramada Hotel	ramada.jpg
5	RadissonBlu Hotel	radissonblu.jpg
6	Raffles Hotel	Raffles-Istanbul-zorlucenter.jpg
7	Hilton İstanbul	hiltonistanbul.jpg
8	SwissOtel	swissotel.jpg
9	Palas Bandırma Hotel	palasBandırma.jpg
10	The Marmara Taksim Hotel	the-marmara-taksim.jpg
11	Sheraton Hotel	sheraton-ankara-jpg
12	Elite World Grand Sapanca	eliteworldgrandsapanca.jpg

Resim 28: Oteller Tablosu

- Sayfanın alt kısmındaki footer bölümü, telif hakkı ve tasarım bilgilerini içerir.
- Sayfanın altında, kullanılan JavaScript dosyalarının bağlantıları bulunmaktadır.

```

<!-- Footer Section -->
<footer id="footer" class="footer">
  <div class="container">
    <div class="copyright">
      &copy; Copyright <strong><span>HotelRate</span></strong>. All Rights Reserved
    </div>
    <div class="credits">
      Designed by HotelRate
    </div>
  </div>
</footer><!-- End Footer -->
<a href="#" class="scroll-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-short"></i></a>
<div id="preloader">
  <div class="line"></div>
</div>
<!-- Vendor JS Files -->
<script src="/wwwroot/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
<script src="/wwwroot/vendor/swiper/swiper-bundle.min.js"></script>
<script src="/wwwroot/vendor/glightbox/js/glightbox.min.js"></script>
<script src="/wwwroot/vendor/aos/aos.js"></script>
<script src="/wwwroot/vendor/php-email-form/validate.js"></script>
<!-- Template Main JS File -->
<script src="/wwwroot/js/main.js"></script>
</body>
</html>

```

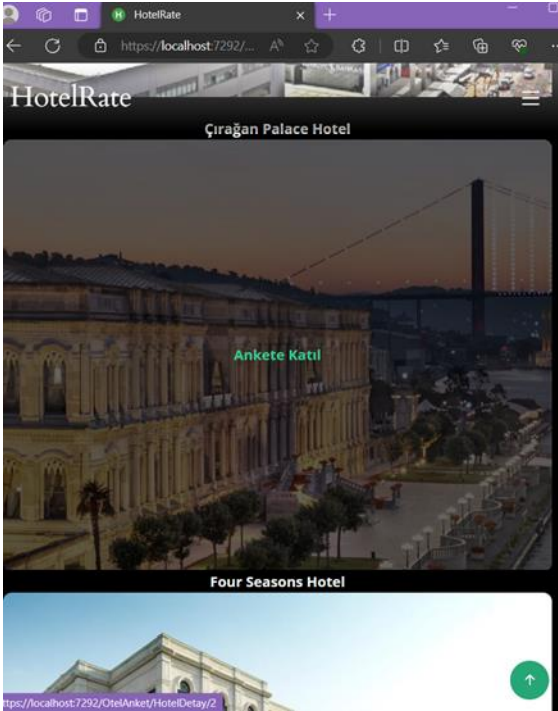
Resim 29: Index View(4)

Anasayfanın Görünümü:



Resim 30: Anasayfanın Görüntüsü

Aynı zamanda sitemiz ekran küçültmelerine uyarlanabilir şekilde tasarlandı.



Resim 31: Anasayfanın küçültülmesi

4.4.7 OTEL ANKET

Bu bölümde kullanıcılar seçtikleri otelin 20 soruluk anketini cevaplayabilir aynı zamanda seçtikleri otel hakkında bilgi alabilirler.

Controller:

Controllerdaki eylemler, kullanıcı etkileşimlerine yanıt olarak belirli işlemleri gerçekleştirir ve ilgili veri tabanı işlemlerini yapar.

- Bu ifadeler, kullandığımız sınıfları ve kütüphaneleri tanımlar. Örneğin, MVC framework'ünü, Entity Framework Core'u ve diğer ilgili özellikleri kullanabilmek için bu ifadeleri ekledik.

```
using HotelRate2.Migrations;  
using HotelRate2.Models;  
using Microsoft.AspNetCore.Mvc;  
using Microsoft.AspNetCore.Mvc.Rendering;  
using Microsoft.Data.SqlClient;  
using Microsoft.EntityFrameworkCore;  
using Microsoft.EntityFrameworkCore.Query;  
using System.Net.WebSockets;
```

Resim 32: OtelAnket Controller(1)

- `public OtelAnketController(HotelDbContext context)` yapıcı metodu, denetleyici sınıfının bir örneği oluşturulduğunda çağrılır ve **HotelDbContext** tipinde bir parametre alır. Bu parametre, veri tabanı işlemlerimiz için kullanılacak olan bir bağlamı temsil eder.

```
namespace HotelRate2.Controllers
{
    1 başvuru
    public class OtelAnketController : Controller
    {
        private readonly HotelDbContext _context;
        0 başvuru
        public OtelAnketController(HotelDbContext context)
        {
            _context = context;
        }
    }
}
```

Resim 33: OtelAnket Controller(2)

- `public IActionResult HotelDetay(int? id)` (GET action), seçilen bir otelin detaylarını ve otele ait soruları veri tabanından çeker.
- Oturumdan kullanıcı ID'sini alır ve View'e iletmek üzere ViewBag kullanır.
- `Tuple<Otel, List<Soru>>` türünde bir Tuple oluşturuldu ve bu tuple görünüme geçirildi.

```
0 başvuru
public IActionResult HotelDetay(int? id)
{
    var otel = _context.Oteller.Include(x => x.OtelResimleri).FirstOrDefault(a => a.OtelId == id);
    var sorular = _context.Sorular.ToList();

    int? userId = HttpContext.Session.GetInt32("KullaniciId");

    ViewBag.SelectedHotelId = otel.OtelId;

    var tuplecreate = Tuple.Create<Otel, List<Soru>>(otel, sorular);

    return View(tuplecreate);
}
```

Resim 34: OtelAnket Controller(2)

- POST action, kullanıcının anket cevaplarını gönderdiği durumu işler.
- Form verilerini kullanarak cevapları çıkarıp ve veri tabanına kaydeder.
- Son olarak, kullanıcıyı "Check" controllerdaki "Kontrol" actionına yönlendirir.

```
[HttpPost]
0 başvuru
public IActionResult HotelDetay(FormCollection form)
{
    var routes = Request.RouteValues;

    int selectedHotelId = int.Parse(routes["id"].ToString());

    int? KullanicId = HttpContext.Session.GetInt32("KullaniciId");

    foreach (var key in form.Keys)
    {
        if (key.StartsWith("SoruId_"))
        {
            int soruId = int.Parse(key.Substring(7));
            string cevap = form[key];

            var cevapModel = new Cevap(soruId, cevap)
            {
                OtelId = selectedHotelId,
                KullanicilarId = KullanicId ?? 0,
            };

            _context.Cevaplar.Add(cevapModel);
            _context.SaveChanges();
        }
    }

    return RedirectToAction("Kontrol", "Check", new { id = selectedHotelId });
}
```

Resim 35: OtelAnket Controller(3)

CustomFormCollectionModelBinderProvider Sınıfı:

- Bu sınıfın temel amacı, bir HTTP isteği içindeki form verilerini doğrudan FormCollection türüne bağlamak ve bu özel model bağlamayı kullanarak model bağlama işlemlerini özelleştirmektir. Özellikle, bu yaklaşım, belirli bir durumda sadece form verilerini almak istediğinizde veya özel bir form verisi işleme mantığı uygulamak istediğinizde kullanılabilir. Bizde otel sorularının cevaplarını kaydederken bu yöntemi kullandık.

```
using Microsoft.AspNetCore.Mvc.ModelBinding;

1 başvuru
internal class CustomFormCollectionModelBinderProvider : IModelBinderProvider
{
    0 başvuru
    public IModelBinder GetBinder(ModelBinderProviderContext context)
    {
        if (context == null)
        {
            throw new ArgumentNullException(nameof(context));
        }

        if (context.Metadata.ModelType == typeof(FormCollection))
        {
            return new CustomFormCollectionModelBinder();
        }

        return null;
    }
}

1 başvuru
public class CustomFormCollectionModelBinder : IModelBinder
{
    0 başvuru
    public Task BindModelAsync(ModelBindingContext bindingContext)
    {
        if (bindingContext == null)
        {
            throw new ArgumentNullException(nameof(bindingContext));
        }

        var formCollection = bindingContext.HttpContext.Request.Form;
        bindingContext.Result = ModelBindingResult.Success(formCollection);
        return Task.CompletedTask;
    }
}
```

Resim 36: OtelAnket Controller(4)

View:

Burada otel anket sayfasının tasarımını ve kullanıcıdan cevap almak için kullanılan bir formu temsil ekledik. Sayfa, kullanıcının her bir soruya bir yanıt seçmesini sağlar ve bu yanıtları "Anketi Kaydet" butonuna tıklayarak sunucuya gönderebilir.

- Burada, ASP.NET Core Tag Helper'ları kullanımını etkinleştirdik ve görünüm, bir Tuple içinde bir otel ve otele ait soruların listesini tanımladık.

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@model Tuple<HotelRate2.Models.Otel, List<HotelRate2.Models.Soru>>
```

Resim 37: HotelDetay View(1)

- Bu blok, sayfa için oluşturduğumuz Layoutu belirler ve formda kullanacağımız bir sayac adında bir değişken tanımlar.

```
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
    int sayac = 0;
}
```

Resim 38: HotelDetay View(2)

- Bu bölüm , seçilen otelin adını veri tabanından çeker ve hoş geldiniz mesajını içerir. Aynı zamanda bu bölümün tasarımında kullanılan bazı html kodlarını da içerir.

```
<div class="page-header d-flex align-items-center">
  <div class="container position-relative">
    <div class="row d-flex justify-content-center">
      <div class="col-lg-6 text-center">
        <h2>@Model.Item1.OtelAdi</h2>
        <p>Merhaba HotelRate'e hoş geldiniz... @Model.Item1.OtelAdi'da konaklama deneyimlerinizi aşağıdaki anket sorularını cevaplayarak oteli değerlendirebilirsiniz.</p>
        <a class="cta-btn" href="#survey1">Hemen Başla</a>
      </div>
    </div>
  </div><!-- End Page Header -->
  <div class="card bg-success text-white m-5">
    <div class="card-body text-center">
      <a href="https://www.bandirnapalas.com/" target="_blank" class="text-white">
        <h5 class="card-title">@Model.Item1.OtelAdi SAYFASINA GİTMEK İÇİN TIKLAYINIZ.</h5>
        <p class="card-text">Buradan daha fazla bilgi alabilirsiniz.</p>
      </a>
    </div>
  </div>
  <!-- ===== Gallery Single Section ===== -->
  <section id="gallery-single" class="gallery-single">
    <div class="container">
      <div class="position-relative h-100 p-3">
        <div class="slides-1 portfolio-details-slider swiper w-40">
          <div class="swiper-wrapper align-items-center">
```

Resim 39: HotelDetay View(3)

- Burada seçilen otele ait resimleri göstermek için bir döngü kullanılmıştır. Seçilen otele göre veri tabanından resimleri çekilir.

```
@foreach (var item in Model.Item1.OtelResimleri)
{
  <div class="swiper-slide">
    
  </div>
}
```

Resim 40: HotelDetay View(4)

- Burada anket sorularını gösteren bir döngü bulunur ve her bir soru için bir form alanı oluşturulmuştur.
- Anket soruları seçilen otelin idsine göre veri tabanından çekilir. Bu şekilde daha dinamik bir yapı elde etmiş oluruz.
- Döngü içinde sayaç her sorudan sonra bir arttırılacak şekilde çalışır.
- Her cevap şikkına ait value değeri bu cevapların puanlamasını belirtir.
- Her soru için bir döngü içinde radio butonlar kullanılarak cevap seçenekleri sunulmuştur.
- Form, "HotelDetay" eylemine yönlendirilmiş ve otel ID'si de bir route parametresi olarak eklenmiştir.
- Anketi kaydet butonu seçilen otelin idsine göre Kontrol sayfasına yönlendirilir.



Soru 1

Otelde konaklarken genel olarak güvende hissettim.

Yanıtınız

- ☐ Mükemmel
- ☐ İyi
- ☐ Orta
- ☒ Yetersiz
- ☐ Çok Kötü

Soru 2

Oteldeki oda kapıları, güvenlik kameraları ve diğer fiziksel güvenlik önlemleri yeterliydi

Yanıtınız

- ☐ Mükemmel
- ☐ İyi
- ☒ Orta
- ☐ Yetersiz
- ☐ Çok Kötü



Soru 19

Otelin tesisleri modern ve günceldi.

Yanıtınız

- ☒ Mükemmel
- ☐ İyi
- ☐ Orta
- ☐ Yetersiz
- ☐ Çok Kötü

Soru 20

Otelin sunmuş olduğu rekreasyon alanları, spor olanakları ve diğer tesis imkanları konusunda memnunum.

Yanıtınız

- ☒ Mükemmel
- ☐ İyi
- ☐ Orta
- ☐ Yetersiz
- ☐ Çok Kötü

Anketi kaydet



Resim 43: Otel Anket sayfasının görüntüsü(2)

	CevapId	Cevabi	Kullanıcılar...	OtelId	Soruld
2	1	1	1	1	1
3	1	1	1	1	1
4	2	1	1	1	2
5	3	1	1	1	3
6	4	1	1	1	4
7	5	1	1	1	5
8	4	1	1	1	6
9	1	1	1	1	1
10	2	1	1	1	2
11	3	1	1	1	3
12	4	1	1	1	4
13	5	1	1	1	5
14	4	1	1	1	6
15	5	1	1	1	7
16	2	1	1	1	8
17	1	1	1	1	9
18	1	1	1	1	10
19	5	1	1	1	11
20	2	1	1	1	13
21	4	1	1	1	14
22	2	1	1	1	15
23	5	1	1	1	16
24	2	1	1	1	18
25	4	1	1	1	19
26	2	1	1	1	20
35	1	11	3	3	1
36	2	11	3	3	2
37	3	11	3	3	3
38	4	11	3	3	4

Cevaplanan sorular kişi, otel ve soruya göre cevaplar tablosunda tutulur.

4.4.8 CHECK SAYFASI

Bu sayfada otel anketlerinden gelen veriler kaydedildikten sonra bir kontrol sayfası gelir.

Controller:

CheckController otel bilgilerini kontrol etmek ve görüntülemek için kullanılır.

- Using ifadeleri, bu controller sınıfında kullanılacak olan sınıfları ve ASP.NET Core MVC'ye ait temel sınıfları içeri aktarır.
- Yapıcı metod, **CheckController** sınıfının bir örneği oluşturulduğunda çağrılır ve **HotelDbContext** tipinde bir parametre alır. Bu parametre, veritabanı işlemleri için kullanılacak olan bir bağlamı temsil eder.

```

using HotelRate2.Models;
using Microsoft.AspNetCore.Mvc;

namespace HotelRate2.Controllers
{
    1 başvuru
    public class CheckController : Controller
    {
        private readonly HotelDbContext _context;
        0 başvuru
        public CheckController(HotelDbContext context)
        {
            _context = context;
        }
    }
}

```

Resim 44: Check Controller(1)

- **Kontrol** adlı action, bir otel ID'si alır ve bu ID'ye sahip bir otel kaydını veritabanından çeker.
- Eğer ID **null** ise veya veritabanında ilgili otel bulunamazsa, **NotFound** döndürülür.
- Bulunan oteli içeren bir liste oluşturulur ve bu liste, "Kontrol" adlı View'a gönderilir.

```
public IActionResult Kontrol(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var otel = _context.Oteller.SingleOrDefault(o => o.OtelId == id);

    if (otel == null)
    {
        return NotFound();
    }

    var otelList = new List<Otel> { otel };

    return View("Kontrol", otelList);
}
```

Resim 45: Check Controller(2)

View:

Bu view, kullanıcıya bir otel anketinin başarıyla tamamlandığını ve bazı yönlendirmeler için seçenekleri içeren bir sayfayı gösterir.

- Bu satırlar, ASP.NET Core Tag Helper'ları kullanımını etkinleştirir ve view, bir otel modelinden otel listesi bekler.
- Döngüde kullanacağımız bir sayaç tanımlanmıştır.

```
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@model List<HotelRate2.Models.Otel>

@{
    int sayac = 0;
}
```

Resim 46: Kontrol View(1)

- HTML sayfasının başlık ve meta etiketleri burada tanımlanmıştır.
- Sayfanın favicon ve Google Fonts'a erişimini sağlayan bağlantılar bulunmaktadır.
- Harici CSS dosyalarına olan bağlantılar bulunur.
- Sayfanın başlık bölümü ve navigasyon menüsü vardır.
- İletişim ve Anasayfaya gidebilmek için tag helperlar kullanılmıştır.

```

<!DOCTYPE html>
<html lang="tr">
<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">
  <title>HotelRate-Survey</title>
  <meta content="siteacikla" name="description">
  <meta content="anahtar kelime gir" name="keywords">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <meta http-equiv="X-UA-Compatible" content="IE=7">
  <!-- Favicons -->
  <link href="/wwwroot/img/bannerIcon.png" rel="icon">
  <!-- Google Fonts -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Open+Sans:ital,wght@0,300;0,400;0,500;0,600;0,700;1,300;1,400;1,600;1,700&family=Inter:ital,wght@0,300;0,400;0,500;0,600;0,700;1,300;1,400;1,600;1,700" rel="stylesheet">
  <!-- Vendor -->
  <link href="/wwwroot/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="/wwwroot/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="/wwwroot/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
  <link href="/wwwroot/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="/wwwroot/vendor/aos/aos.css" rel="stylesheet">
  <!-- Template Main CSS File -->
  <link href="/wwwroot/css/main.css" rel="stylesheet">
</head>
<body>
  <!-- ===== Header ===== -->
  <div id="header" class="header d-flex align-items-center fixed-top">
    <div class="container-fluid d-flex align-items-center justify-content-between">
      <a href="index.html" class="logo d-flex align-items-center me-auto me-lg-0">
        
        <h1>HotelRate</h1>
      </a>
      <nav id="navbar" class="navbar px-4">
        <ul>
          <li><a asp-controller="Anasayfa" asp-action="Index">Anasayfa</a></li>
          <li><a asp-controller="Iletisim" asp-action="Index">İletişim</a></li>
        </ul>
      </nav><!-- .navbar -->
    </div>
  </div>
</header><!-- End Header -->
  <main id="main" data-aos="fade" data-aos-delay="1500">

```

Resim 47: Kontrol View(2)

- Sayfanın ana içeriği burada başlar.
- Döngüde otel idsine göre kontrol sayfası döndürülür.
- Anasayfa ve istatistik sayfasında gidebilmek için tag helperler kullanılmıştır.

```

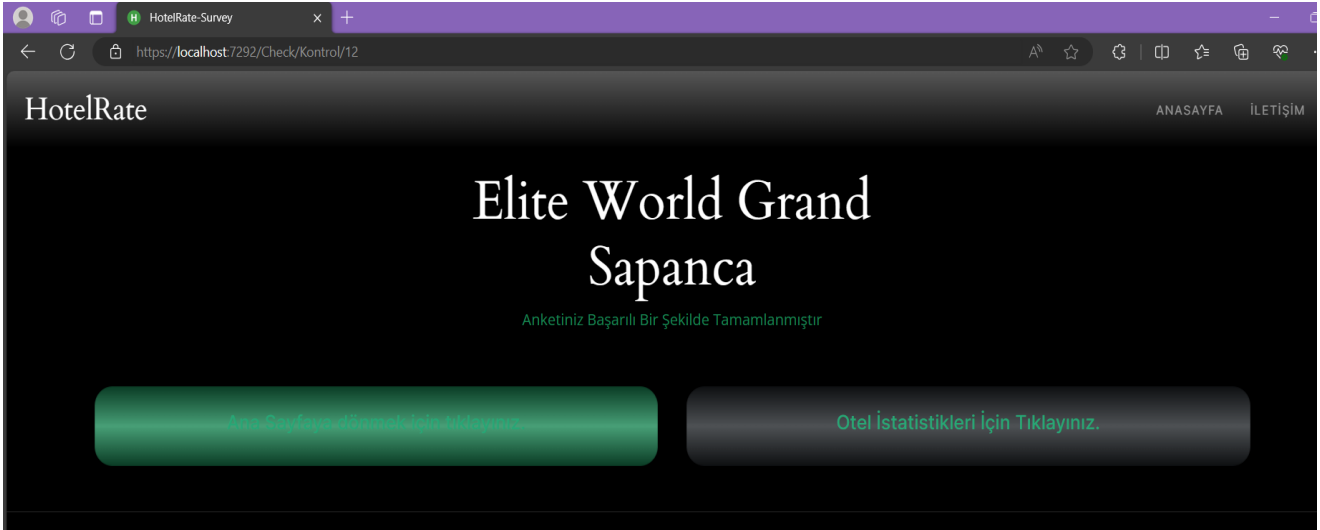
<main id="main" data-aos="fade" data-aos-delay="1500">
  <!-- ===== End Page Header ===== -->
  <div class="page-header d-flex align-items-center">
    @for (int i = 0; i < Model.Count; i++)
    {
      <div class="container position-relative">
        <div class="row d-flex justify-content-center">
          <div class="col-lg-6 text-center">
            <h2>@Model[i].OtelAdi</h2>
            <p class="text-success">Anketiniz Başarılı Bir Şekilde Tamamlanmıştır</p>
            
          </div>
        </div>
      </div>
    }
  </div><!-- End Page Header -->
  <div class="row align-items-center justify-content-center">
    <div class="card bg-success text-white col-lg-5 m-3 p-2">
      <div class="card-body text-center">
        <a asp-controller="Anasayfa" asp-action="Index">
          <h5 class="card-title">Ana Sayfaya dönmek için tıklayınız.</h5>
        </a>
      </div>
    </div>
    <div class="card bg-dark text-white col-lg-5 m-3 p-2">
      <div class="card-body text-center">
        <a asp-controller="Istatistik" asp-action="Index">
          <h5 class="card-title">Otel İstatistikleri İçin Tıklayınız.</h5>
        </a>
      </div>
    </div>
  </div>
</div>

```

Resim 48: Kontrol View(2)

Kontrol sayfasının görünümü:

Aşağıda otel Idsi 12 olan otele ait kontrol sayfası gösterilmiştir.



Resim 49: Kontrol Sayfasının Görünümü

4.4.9 İLETİŞİM SAYFASI

Controller:

```
using Microsoft.AspNetCore.Mvc;

namespace HotelRate2.Controllers
{
    0 başvuru
    public class IletisimController : Controller
    {
        0 başvuru
        public IActionResult Index()
        {
            return View();
        }
    }
}
```

Resim 50: İletisim Controller

View:

```
</div>
</div>
<!-- End Info Item -->
<div class="col-lg-3">
  <div class="info-item d-flex">
    <i class="bi bi-envelope flex-shrink-0"></i>
    <div>
      <h4>E-mail:</h4>
      <p>Email gir</p>
    </div>
  </div>
</div>
<!-- End Info Item -->
<div class="col-lg-3">
  <div class="info-item d-flex">
    <i class="bi bi-phone flex-shrink-0"></i>
    <div>
      <h4>Telefon Numarası:</h4>
      <p>no gir</p>
    </div>
  </div>
</div>
<!-- End Info Item -->
</div>
<div class="row justify-content-center mt-4">
  <div class="col-lg-9">
    <form action="forms/contact.php" method="post" role="form" class="php-email-form">
      <div class="row">
        <div class="col-md-6 form-group">
          <input type="text" name="name" class="form-control" id="name" placeholder="Adınız-Soyadınız" required>
        </div>
        <div class="col-md-6 form-group mt-3 mt-md-0">
          <input type="email" class="form-control" name="email" id="email" placeholder="Mail Adresiniz" required>
        </div>
      </div>
      <div class="form-group mt-3">
        <input type="text" class="form-control" name="subject" id="subject" placeholder="Konu" required>
      </div>
      <div class="form-group mt-3">
        <textarea class="form-control" name="message" rows="5" placeholder="Mesaj" required></textarea>
      </div>
      <div class="my-3">
        <div class="loading">Loading</div>
        <div class="error-message"></div>
        <div class="sent-message">Mesajın gönderildi. Teşekkürler!</div>
      </div>
      <div class="text-center"><button type="submit">Mesaj Gönder</button></div>
    </form>
  </div>
</div>
</div>
<!-- End Contact Section -->
<!-- End #main -->
===== Footer ===== -->
<div class="Footer">
```

Resim 51: İletişim View

İletişim Sayfasının Görünümü:

H

HotelRate

ANASAYFA İLETİŞİM

İletişim

Konum:

Konum yaz

E-mail:

Email gir

Telefon Numarası:

no gir

Adınız-Soyadınız

Mail Adresiniz

Konu

Mesaj

Mesaj Gönder

© Copyright Hotel. All Rights Reserved
Designed by HotelRate

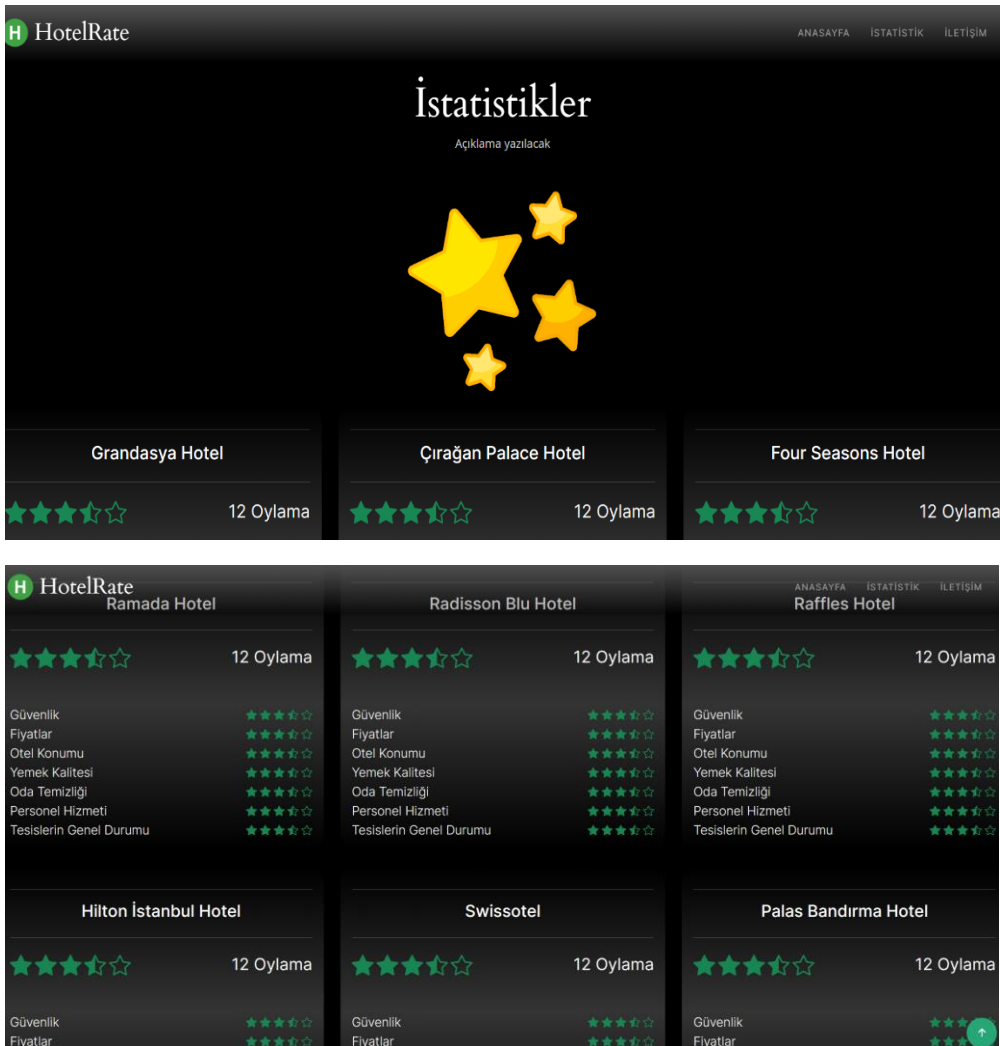
Resim 52: İletişim Sayfasının Görünümü

4.5 GELİŞTİRİLECEK BÖLÜMLER

HotelRate projemiz ileri süreçteki geliştirme ve yeniliklere uyumlu olarak tasarlandı.

- Mobil cihazlarda kullanımı kolaylaştırılacak, çünkü birçok kullanıcı mobil cihazlardan anketlere erişiyor.
- Farklı kategorilerde ve konularda anketler oluşturularak daha geniş bir katılımcı kitlesi çekmeyi hedefliyoruz.
- Sitemiz dinamik bir yapıda olmasından dolayı otel sayısını arttırarak daha geniş bir anket sistemi oluşturmayı planlıyoruz.
- Kullanıcıların profillerini oluşturulacak. Bu sayede kullanıcılar daha önce katıldıkları anketleri görebilecek ve gelecekteki anket önerileri alabilecekler.
- Katılımcıların verilerini korumak adına daha güçlü bir güvenlik politikası uygulanacak.
- Kullanıcılara geri bildirimlerini iletebilecekleri bir iletişim kanalı sağlanacak.
- Bir yardımcı chat oluşturulabilir.
- Son olarak şuan sitemizde demo olarak gösterilen ve geliştirme aşamasında olan istatistik sayfası anket sonuçlarını konulara ayırarak istatistiklerini analiz etmeyi hedeflemektedir. Anket sonuçlarından elde edilen verileri anlamlı ve etkili bir şekilde raporlanacak.

İstatistik Sayfası Demosu:



Resim 52: İstatistik Sayfasının Demosu

5. TARTIŞMA VE SONUÇ

Raporda gösterilen önceki çalışmalar ve bizim projemiz arasında gözle görülür farklar vardır.

Kullanıcı Arayüzü:

- Önceki Anket Sistemleri: Genellikle temel arayüz ve sınırlı özelliklere sahiptir. Kullanıcı deneyimi daha sadedir.
- HotelRate Anket Sistemi: Kullanıcı dostu, etkileşimli ve modern arayüzlerle donatılmıştır. Çeşitli özellikler sunar.

Özellik Çeşitliliği:

- Önceki Anket Sistemleri: Genellikle temel anket oluşturma ve yanıt toplama özelliklerini içerir. Ve tek bir seçenek üzerinden ilerlenir.
- HotelRate Anket Sistemi: Çoklu soru tipleri, analiz araçları, çoklu anket seçeneği sunar.

Mobil Uyum:

- Önceki Anket Sistemleri: Mobil uyum konusunda sınırlıdır, mobil cihazlarda kullanıcı dostu olmayabilir.
- Önceki Anket Sistemleri: Tam mobil uyum sağlar ve kullanıcıların anketlere kolayca mobil cihazlar üzerinden erişmelerine imkan tanır.

6. KAYNAKÇA

- [1] <https://ankara.onturhotels.com/memnuniyet-anketi>
- [2] <http://anket.anketix.com/fs.aspx?surveyid=2c549c4879847109bec10b4037e7758>
- [3] <https://getbootstrap.com/>
- [4] <https://www.questionpro.com/blog/tr/web-anketleri-nedir-orneklerle-tanim-yontemler-ve-turler/>
- [5] <https://tr.wikipedia.org/wiki/Anket>

KATKI

BEYZA ATAY – Site Tasarımı, BackEnd, Veritabanı Sistemi, Rapor(Giriş,Literatür Taraması,Projeyi kodlarla beraber açıklama,Geliştirme Bölümü), Sunum

MERVE BAĞIŞLAR Site Tasarımı, BackEnd, Veritabanı Sistemi, Rapor(Giriş,Literatür Taraması,Projeyi kodlarla beraber açıklama,Geliştirme Bölümü), Sunum

