



T.C.

SAKARYA ÜNİVERSİTESİ

WEB PROGRAMLAMA PROJE RAPORU

G201210557 - OKTAY TÜRKMEN

GRUP :2B

GİTHUP: <https://github.com/oktaytr1/WebProgramlamaProje>

1.Proje Tanıtımı

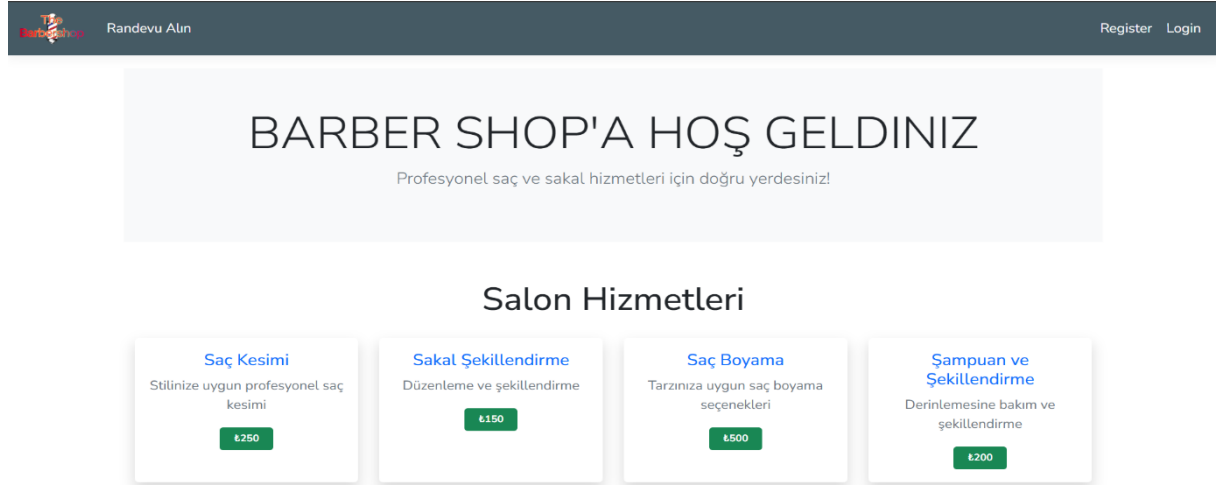
Bu proje, Web Programlama dersi kapsamında geliştirilmiş bir Kuaför uygulaması olup, modern web teknolojilerini kullanarak kullanıcı dostu, güvenli ve performanslı bir web tabanlı uygulama oluşturmayı hedeflemektedir.

amaçlar

- Kullanıcı dostu bir arayüz tasarımı ile kolay, hızlı ve sezgisel bir deneyim sunmak.
- Etkin ve ölçeklenebilir bir veritabanı yapısı kullanarak, verilerin güvenli ve tutarlı bir şekilde saklanması ve yönetimini sağlamak.
- En yeni ve güncel web teknolojilerinden yararlanarak güvenlik, performans ve erişilebilirlik standartlarını üst düzeyde tutan bir uygulama geliştirmek.


Proje, kullanıcıların ihtiyaçlarını hızlı ve etkin bir şekilde karşılamayı amaçlarken aynı zamanda geliştirici dostu bir altyapı sunmayı hedeflemektedir.

BarberShop Ana Ekran



Kayıt Ol Sayfası

Yeni kayıt olmak isteyen kullanıcıların kayıt olma sayfası. Önceden kayıtlı E-postaların kontrol edildiği veritabanından kayıtlı e-posta varsa bu e-posta zaten kayıtlı kontrollerinin yapıldığı sayfa.

Randevu Alın

Register Login

Register

Create a new account.

Register

Use another service to register.

There are no external authentication services configured. See this [article](#) about setting up this ASP.NET application to support logging in via external services.

© 2024 - OKTAY TÜRKMEN - [Privacy](#) [View Source](#)

Kayıtlı kullanıcı Giriş Sayfası

Sisteme kayıtlı kullanıcıların randevu alabildiği, önceden aldığı randevuları sorgulayabildiği ve yapay zekâ desteğiyle saç önerisi alabildiği sayfa.

Randevu Alın Saç analizi Randevularım

Hello oktaytr@gmail.com! Logout

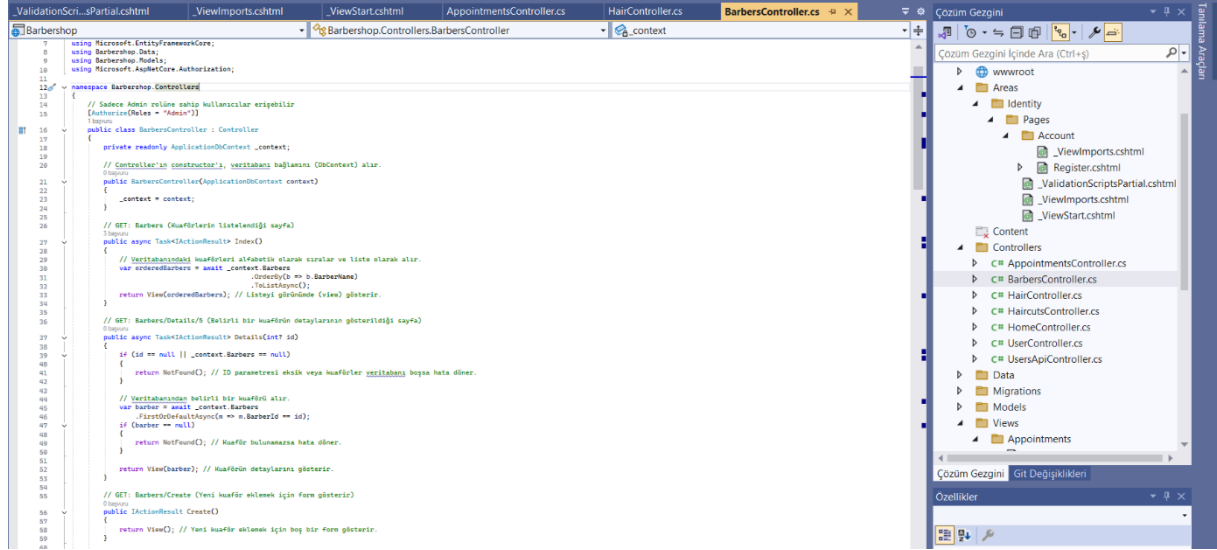
Merhaba, oktaytr@gmail.com!

Sistemimizde keyifli vakit geçirmenizi dileriz.

Randevu Al

LINQ kullanım Örneği

Projenin bir çok kısmında özellikle controller kısımlarında LINQ kullanımıyla alan veri tabanındaki uygun verilerden sorgulamalar yapıldı.



RESTAPI kullanımı

Kullanılan Yöntemler

REST (Representational State Transfer), istemci ile sunucu arasındaki iletişimi HTTP protokolü üzerinden düzenleyen bir mimaridir. Proje, kullanıcı yönetimini aşağıdaki yollarla gerçekleştirir:

- **HTTP GET:** Kullanıcı listesini istemek için kullanılır.
- **HTTP DELETE:** Belirli bir kullanıcıyı silmek için kullanılır.

1. Kullanıcı Listeleme:

- **Endpoint:** /api/userapi
- **HTTP Metodu:** GET
- Veritabanındaki kullanıcılar JSON formatında döndürülür ve frontend'de tabloya eklenir.

2. Kullanıcı Silme:

- **Endpoint:** /api/userapi/{id}
- **HTTP Metodu:** DELETE

```
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;

namespace Barbershop.Controllers
{
    [Route("api/[controller]")] // API'nin route'u "/api/userapi" olarak tanımlanır
    [ApiController]
    1 başvuru
    public class UserApiController : ControllerBase
    {
        private readonly UserManager<IdentityUser> _userManager; // Kullanıcı yönetimi için bağımlılık

        0 başvuru
        public UserApiController(UserManager<IdentityUser> userManager)
        {
            _userManager = userManager; // Bağımlılığı sınıfa atar
        }

        // Kullanıcıların listesini döndüren endpoint
        [HttpGet]
        0 başvuru
        public IActionResult GetUsers()
        {
            // Tüm kullanıcıları listele ve yalnızca gerekli bilgileri seç
            var users = _userManager.Users.Select(u => new
            {
                u.Id,           // Kullanıcı ID'si
                u.UserName,     // Kullanıcı adı
                u.Email         // E-posta adresi
            }).ToList();

            return Ok(users); // Kullanıcıları JSON formatında döndür
        }

        // Belirli bir kullanıcıyı silen endpoint
        [HttpDelete("{id}")]
        0 başvuru
        public async Task<IActionResult> DeleteUser(string id)
        {
            // Verilen ID'ye sahip kullanıcıyı bul
            var user = await _userManager.FindByIdAsync(id);
            if (user == null)
            {
                return NotFound(new { Message = "User not found" }); // Kullanıcı bulunamazsa hata döndür
            }

            // Kullanıcıyı sil
            await _userManager.DeleteAsync(user);
            return Ok(new { Message = "User deleted successfully" }); // Başarı mesajı döndür
        }
    }
}
```

Front-End ile API İletişimi:

Kullanıcılar, JavaScript kullanılarak REST API aracılığıyla alınmakta ve tabloya dinamik olarak eklenmektedir.

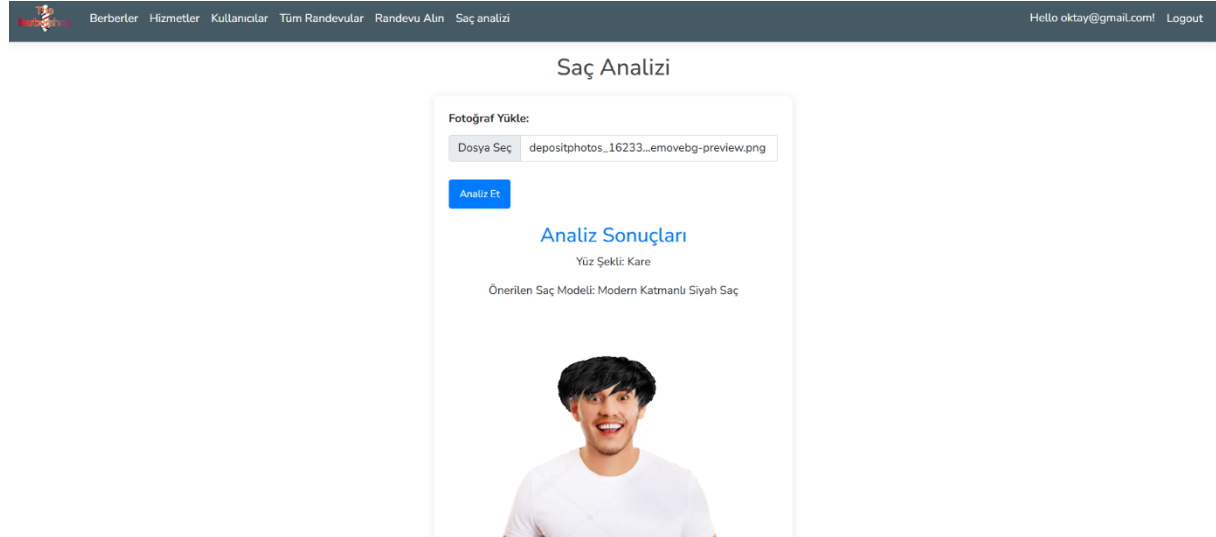
Yapay Zekâ Entegrasyonu

Bu Flask uygulaması, yüz şekline uygun saç modeli öneren ve önerilen modeli yüz üzerine bindiren bir yapay zekâ sistemi olarak çalışmaktadır.

- Fotoğrafi işler ve yüz şekli analiz eder.
- Önerilen saç modelini seçer.
- Seçilen modeli fotoğrafa bindirir.
- Çıktıyı bir dosyada saklar ve kullanıcıya sonucu JSON olarak döner

Kullanılan Teknolojiler **Flask**: Web uygulamasını çalıştırır ve API endpointlerini sağlar.

Mediapipe: Yüz analizi ve yüz özelliklerini belirleme işlemleri için kullanılır. **OpenCV**: Görsel işleme (saç modelini yüz üzerine bindirme) işlemleri için kullanılır.



VERİ TABANI MODELİ

Veritabanı Yapısı: Proje, aşağıdaki veritabanı tablolarından oluşmaktadır Ve MYSQL kullanılmıştır

1. Randevular Tablosu (Appointments Table)

Alan Adı	Açıklama	Tür	Anahtar
AppointmentId	Randevunun benzersiz kimliği	int	Primary Key
AppointmentDateTime	Randevunun tarihi ve saati	DateTime	
BarberId	Randevuyu gerçekleştirecek berberin kimliği	int	Foreign Key
HaircutId	Randevuda seçilen saç kesimi kimliği	int	Foreign Key
CustomerId	Randevuyu alan müşterinin kimliği	string	Foreign Key
Status	Randevunun durumu	AppointmentStatus	

İlişkiler:

- Barber (Foreign Key):** Randevunun bağlı olduğu berber.
 - Haircut (Foreign Key):** Randevunun seçilen hizmeti.
 - User (IdentityUser Foreign Key):** Randevuyu alan müşteri.
-

2. Berberler Tablosu (Barbers Table)

Alan Adı	Açıklama	Tür	Anahtar
BarberId	Berberin benzersiz kimliği	int	Primary Key
BarberName	Berberin adı	string	
Specialization	Berberin uzmanlık alanı	string	

İlişkiler:

- **Appointments (1:N):** Berberin aldığı tüm randevular.

3. Hizmetler Tablosu (Haircuts Table)

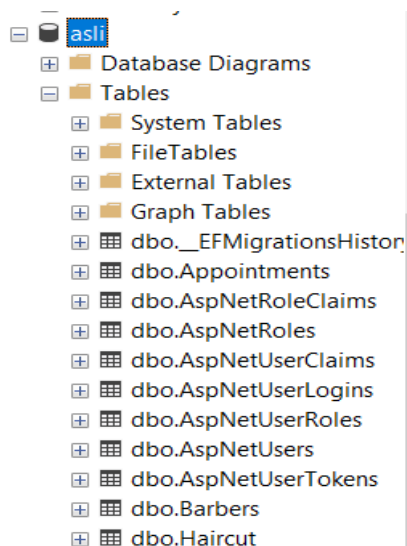
Alan Adı	Açıklama	Tür	Anahtar
HaircutId	Hizmetin benzersiz kimliği	int	Primary Key
Name	Hizmetin adı	string	
Price	Hizmetin fiyatı	double	
Duration	Hizmetin süresi (dakika)	int	

İlişkiler:

- **Appointments (1:N):** Hizmetin seçildiği tüm randevular.

İlişkiler Tablosu Özet:

Veri tabanı Şeması



Sonuç

Bu proje, Web Programlama dersi kapsamında geliştirilerek hem teknik becerilerin geliştirilmesi hem de modern web teknolojilerinin pratikte uygulanması amacıyla tasarlanmıştır, dokümanda istenilen bütün gereksinimler karşılanmıştır ve hepsi doğru bir şekilde çalışmaktadır. Projenin tamamlanmasıyla kullanışı kolay ve sade bir web uygulaması ortaya çıkarılmıştır. Veritabanı yapısının etkin bir şekilde kullanılması ve uygulamanın ekran tasarımları, projeyi bir bütün haline getirmiştir.