



**softITo 3. DÖNEM 3. GRUP
BACKEND
BİTİRME PROJESİ RAPORU**

**SmartGymSystem
Akıllı Spor Salonu Yönetim Sistemi**

Berkay Eldut

Temmuz, 2026
softITo Yazılım Bilişim Akademisi, İstanbul

İÇİNDEKİLER

1. GİRİŞ	1
1.1 Projenin Amacı	1
1.2 Projenin Kapsamı	1
2. MATERYAL VE YÖNTEM	2
2.1 Kullanılan Teknolojiler	2
2.2 Veri Erişimi ve Tasarım Desenleri	2
3. SİSTEM MİMARİSİ VE VERİTABANI	3
3.1 Katmanlı Mimari	3
3.3 Veritabanı İlişkileri	4
4. UYGULAMA MODÜLLERİ	5
5. KURULUM VE ÇALIŞTIRMA	6
6. UYGULAMA EKRANLARI	7
7. TEST, GÜVENLİK VE PERFORMANS	19
8. SONUÇ VE GELİŞTİRME ÖNERİLERİ	20
9. KAYNAKÇA	20

ÖZET

SmartGymSystem, spor salonunun üyelik, antrenör, antrenman, randevu, grup dersi, ürün, sipariş ve raporlama süreçlerini tek bir web uygulamasında birleştiren ASP.NET Core MVC tabanlı bir yönetim sistemidir. Uygulama üç temel kullanıcı rolü üzerine kurulmuştur: yönetici, antrenör ve üye. Her rol yalnızca kendi yetkilerine uygun ekranlara ve işlemlere erişebilmektedir.

Projenin backend tarafında ASP.NET Core 9.0, Entity Framework Core, SQL Server, Repository ve Unit of Work desenleri, Dapper, Cookie ve JWT kimlik doğrulama, MemoryCache, Serilog, ClosedXML ve QuestPDF kullanılmıştır. Chatbot modülü yapılandırma mevcutsa Gemini API ile çalışmakta; API anahtarı bulunmadığında yerel kural tabanlı cevap sistemine dönmektedir.

Uygulama, yalnızca klasik bir spor salonu otomasyonu değil; üyeye özel hedef takibi, vücut ölçümü kaydı, akıllı antrenman programı üretimi, antrenör-üye eşleştirmesi, e-ticaret ve yönetim raporlamasını aynı sistem içerisinde sunan bütünsel bir çözümdür.

Anahtar kelimeler

ASP.NET Core MVC, C#, SQL Server, Entity Framework Core, Dapper, Repository, Unit of Work, JWT, Cookie Authentication, SmartGym, spor salonu otomasyonu.

1. GİRİŞ

1.1 Projenin Amacı

Spor salonlarında üyelik bilgilerinin, antrenör atamalarının, grup derslerinin, ödeme ve mağaza süreçlerinin farklı araçlarda takip edilmesi; veri tekrarına, zaman kaybına ve yetki karmaşasına neden olabilmektedir. SmartGymSystem projesinin amacı bu işlemleri merkezi, rol tabanlı ve genişletilebilir bir web uygulaması altında toplamaktır.

Sistem; yöneticinin operasyonu izlemesini, antrenörün kendisine atanan üyeleri takip etmesini ve üyenin antrenman, randevu, üyelik ve alışveriş işlemlerini çevrim içi gerçekleştirmesini hedeflemektedir. Bunun yanında, hedefe göre otomatik antrenman programı oluşturma ve chatbot desteğiyle kullanıcı deneyiminin geliştirilmesi amaçlanmıştır.

1.2 Projenin Kapsamı

- Kullanıcı kaydı, giriş, çıkış ve rol bazlı yönlendirme.
- Yönetici tarafından üye, antrenör, paket, grup dersi, ürün, sipariş ve ödeme yönetimi.

- Üye panelinde üyelik durumu, vücut ölçümleri, randevular ve antrenman programlarının takibi.
- Antrenör panelinde atanmış üyelerin gelişim kayıtları ve programlarının yönetimi.
- Ürün kataloğu, kategori filtreleme, sepet, stok kontrolü, ödeme ve sipariş takibi.
- Hedef bazlı akıllı antrenman programı üretimi.
- Excel ve PDF rapor çıktıları, yönetim raporları ve uygulama logları.
- MVC ekranlarının yanında JWT kullanan REST API uç noktaları.

1.3 Projenin Kullanıcı Roller

Rol	Temel Yetkiler	Örnek İşlemler
Yönetici	Tüm operasyonel verileri yönetir.	Üye/antrenör ekleme, paket ve ürün yönetimi, sipariş durumu, raporlama.
Antrenör	Kendisine atanan üyelerle sınırlı çalışır.	Üye gelişimi, antrenman programı hazırlama, randevu onayı.
Üye	Kendi profil ve işlemlerine erişir.	Ölçüm güncelleme, derse katılma, randevu, program, alışveriş.

2. MATERYAL VE YÖNTEM

2.1 Kullanılan Teknolojiler

Teknoloji	Projede Kullanım Amacı
ASP.NET Core MVC (.NET 9)	Web uygulamasının controller, view ve yönlendirme altyapısı.
C#	Backend iş kuralları, veri modelleri ve servislerin geliştirilmesi.
Razor Views / Bootstrap / JavaScript	Sunucu taraflı HTML üretimi, responsive tasarım ve istemci etkileşimi.
SQL Server	Üyeler, antrenörler, paketler, siparişler ve diğer ilişkisel verilerin saklanması.
Entity Framework Core	Nesne-ilişkisel eşleme, LINQ sorguları ve ilişkilerin yönetimi.
Dapper	Üye listeleme API'sinde yüksek performanslı doğrudan SQL sorgusu.
Repository + Unit of Work	Tekrarlanan veri erişim kodlarını soyutlamak ve kayıt işlemlerini tek noktadan tamamlamak.
Cookie Authentication	MVC kullanıcı oturumlarının yönetimi.
JWT Bearer	REST API istemcileri için token tabanlı kimlik doğrulama altyapısı.
MemoryCache	Sık kullanılan üye listesini kısa süreli önbelleğe alma.
Serilog	Konsol ve günlük dosyasına yapılandırılmış log yazma.
ClosedXML / QuestPDF	Üye listesini Excel ve PDF olarak dışa aktarma.
Gemini API	Chatbot modülünde yapay zekâ destekli cevap üretimi; bağlantı yoksa yerel cevap sistemi.

2.2 ASP.NET Core MVC Yaklaşımı

MVC mimarisi uygulamayı Model, View ve Controller sorumluluklarına ayırır. Model sınıfları veritabanı varlıklarını, View dosyaları kullanıcı arayüzünü, Controller sınıfları ise HTTP istekleriyle iş kuralları arasındaki koordinasyonu temsil etmektedir. Bu ayırım kodun okunabilirliğini ve modüllerin birbirinden bağımsız geliştirilmesini kolaylaştırır.

```
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
```

Varsayılan rota, uygulama ilk açıldığında HomeController içindeki Index eylemine gider. Diğer ekranlar controller ve action adlarına göre yönlendirilir.

2.3 Veri Erişimi ve Tasarım Desenleri

SmartGymSystem veri erişiminde hibrit bir yaklaşım kullanmaktadır. Ana MVC modüllerinde Entity Framework Core ile ilişkisel sorgular ve değişiklik takibi yapılırken, API tarafındaki toplu üye sorgusunda Dapper kullanılmaktadır. Generic Repository temel CRUD işlemlerini soyutlar; Unit of Work ise aynı DbContext üzerinden yapılan değişiklikleri CompleteAsync metodu ile kaydeder.

```

public IRepository<T> Repository<T>() where T : class
{
    var type = typeof(T).Name;
    return (IRepository<T>)_repositories.GetOrAdd(
        type, _ => new Repository<T>(_context));
}

```

Bu yapı, belirli bir model için repository örneğinin tekrar kullanılmasını ve veri erişim bağımlılıklarının controller dışında yönetilebilmesini sağlar.

2.4 Kimlik Doğrulama, Yetkilendirme ve Yardımcı Servisler

MVC ekranlarında Cookie Authentication, API girişinde ise JWT Bearer altyapısı yapılandırılmıştır. Kullanıcı girişinde ad, rol, kullanıcı kimliği ve role özel profil kimliği claim olarak oluşturulur. AdminController, TrainerController ve MemberController sınıflarında rol bazlı Authorize nitelikleriyle ekran erişimleri sınırlandırılmıştır.

```

[Authorize(Roles = "Admin")]
public class AdminController : Controller
{
    // Sadece Admin rolündeki kullanıcılar erişebilir.
}

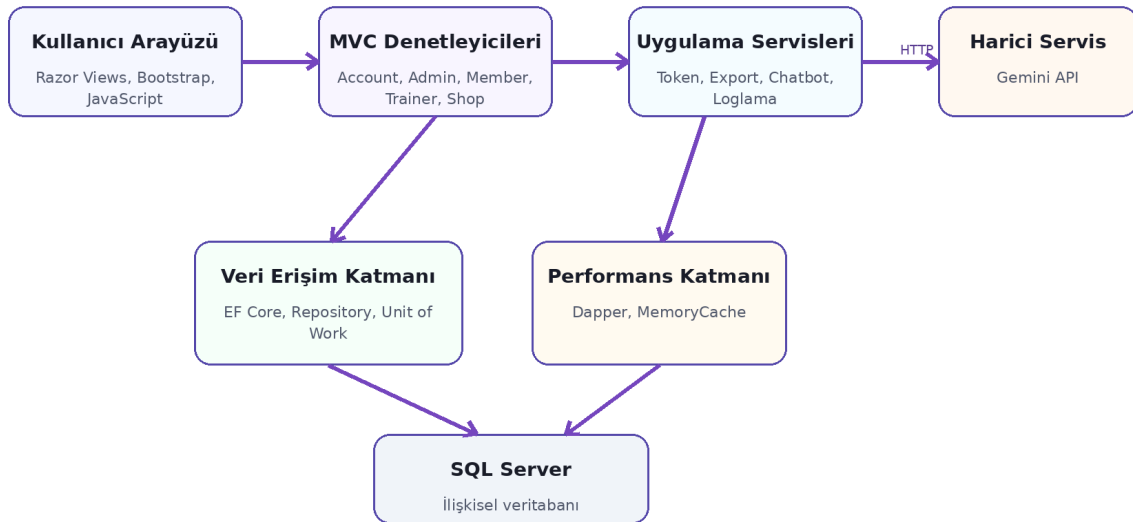
```

ExportService ClosedXML ile Excel, QuestPDF ile PDF çıktısı üretir. ChatbotService, Gemini API anahtarı tanımlıysa dış servise istek gönderir; anahtar yoksa üyelik, antrenman, beslenme ve çalışma saatleri için yerel kural tabanlı cevaplar döndürür. Serilog ise uygulama olaylarını konsola ve günlük dosyalarına kaydeder.

3. SİSTEM MİMARİSİ VE VERİTABANI

3.1 Katmanlı Mimari

SmartGymSystem Katmanlı Mimari



Şekil 1. SmartGymSystem katmanlı mimarisi

Kullanıcı istekleri Razor arayüzünden controller katmanına ulaşır. Controller'lar iş akışını servisler ve veri erişim bileşenleriyle yürütür. Veriler SQL Server'da saklanırken chatbot gerektiğinde Gemini API ile haberleşir.

3.2 Proje Klasör Yapısı

Controllers: HTTP isteklerini karşılayan Account, Admin, Member, Trainer, Shop, Chatbot ve API controller sınıfları.

Models: Veritabanı varlıkları ve form/rapor verilerini taşıyan ViewModel sınıfları.

Data: DbContext, başlangıç verileri, Repository, Dapper ve Unit of Work bileşenleri.

Helpers: Şifre, token, chatbot ve dışa aktarma servisleri.

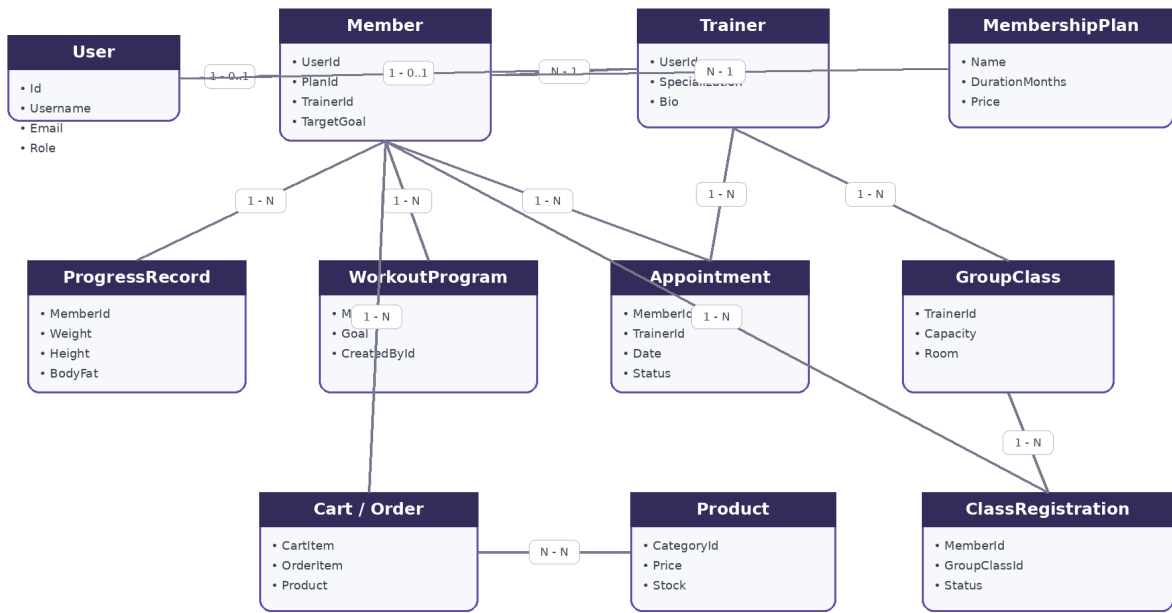
Views: Razor tabanlı kullanıcı, antrenör ve yönetici ekranları.

wwwroot: CSS, JavaScript, ürün ve egzersiz görselleri gibi statik dosyalar.

Logs: Serilog tarafından üretilen günlük dosyaları.

3.3 Veritabanı İlişkileri

Temel Veritabanı İlişkileri



Şekil 2. Temel veritabanı ilişkileri

SmartGymDbContext, kullanıcı profillerinden mağaza siparişlerine kadar 18 DbSet içerir. İlişkiler OnModelCreating metodu içinde Fluent API ile tanımlanmıştır.

User tablosu ortak kimlik bilgilerini tutar ve Member ya da Trainer profiliyle bire bir ilişki kurar. Member; üyelik planı ve atanmış antrenörle ilişkilidir. Üyenin gelişim kayıtları, randevuları, antrenman programları, sepeti, siparişleri ve grup dersi kayıtları ayrı tablolarda tutulur. Bu yapı veri tekrarını azaltır ve her iş alanının bağımsız yönetilmesini sağlar.

3.4 Silme Davranışları ve Veri Bütünlüğü

Üye ile antrenör arasındaki ilişkide ve antrenöre bağlı randevu/grup dersi ilişkilerinde Restrict silme davranışı kullanılmıştır. Böylece bağlı kayıtlar varken antrenörün yanlışlıkla silinmesi önlenir. Üyeye ait program ve randevularda ise yaşam döngüsü üyeye bağlı olduğu için Cascade davranışı tercih edilmiştir. Para ve ölçüm alanları için decimal(18,2) kolon tipi tanımlanarak sayısal hassasiyet korunmuştur.

4. UYGULAMA MODÜLLERİ

4.1 Hesap ve Oturum Yönetimi

AccountController; giriş, kayıt, çıkış ve erişim reddi işlemlerini yürütür. Yeni kayıt olan kullanıcı Member rolüyle oluşturulur; buna bağlı Member profili ve alışveriş sepeti otomatik olarak eklenir. Başarılı giriş sonrasında kullanıcı rolüne göre Admin, Trainer veya Member paneline yönlendirilir.

4.2 Üye Modülü

- Üyelik paketi ve kalan süre bilgisi.
- Atanmış antrenör ve hedef görüntüleme.
- Kilo, boy, yağ oranı ve çevre ölçümlerini kaydetme.
- Grup derslerine katılma ve kaydı iptal etme.
- Antrenörden özel randevu talep etme.
- Hazırlanan antrenman programlarını ve detaylarını görüntüleme.
- Hedef seçerek akıllı program oluşturma.

ApplyRecommendation eylemi, hedefe göre program başlığını oluşturur ve DbInitializer içindeki egzersiz havuzundan uygun hareketleri gün, set, tekrar, dinlenme ve not bilgileriyle programa ekler. Bu yöntem makine öğrenmesi modeli yerine açıklanabilir, kural tabanlı bir eşleştirme yaklaşımıdır.

4.3 Antrenör Modülü

TrainerController yalnızca Trainer rolüne açıktır. Antrenör kendisine atanmış üyeleri görüntüler, üyelerin ölçüm geçmişine ve programlarına erişir, seçilen egzersizlerden kişiye özel program oluşturur ve gelen randevu taleplerinin durumunu günceller. Sorgularda AssignedTrainerId kontrolü yapılarak antrenörün başka bir antrenöre ait üyelere erişmesi engellenir.

4.4 Yönetici Modülü

AdminController sistemin en kapsamlı modülüdür. Dashboard ekranında toplam üye, aktif üyelik, ödeme, ürün, düşük stok ve sipariş göstergeleri hazırlanır. Üye, antrenör, üyelik paketi, grup dersi ve ürünler için ekleme, listeleme, güncelleme ve silme işlemleri bulunur. Siparişlerin durumları yönetilir, ödeme kayıtları görüntülenir ve raporlama ekranı oluşturulur.

4.5 Spor Mağazası ve Sipariş Süreci

Mağaza kataloğu oturum açmadan görüntülenebilir; sepete ekleme, ödeme ve sipariş geçmişi ise Member rolüne özeldir. Sepete ekleme ve ödeme öncesinde stok kontrolü yapılır. Checkout aşamasında toplam tutar hesaplanır, Order ve OrderItem kayıtları oluşturulur, ürün stokları azaltılır ve sepet temizlenir. Üye daha sonra siparişlerinin durumunu takip edebilir.

4.6 API, Önbellek ve Dışa Aktarma

ApiAuthController kullanıcı adı ve şifre doğrulandıktan sonra dört saat geçerli JWT üretir. ApiMembersController üye listesini Dapper ile sorgular ve sonucu beş dakikalık mutlak, iki dakikalık kayan süreyle MemoryCache içerisinde tutar. Yeni üye oluşturulduğunda önbellek temizlenir. Aynı controller, üye listesini ClosedXML ile XLSX ve QuestPDF ile PDF biçiminde dışa aktarır.

```
if (!_memoryCache.TryGetValue(MembersCacheKey, out members))
{
    members = await _dapperRepository.QueryAsync<dynamic>(sql);
    _memoryCache.Set(MembersCacheKey, members, cacheOptions);
}
```

4.7 Chatbot Modülü

Sayfaların sağ alt bölümünde erişilebilen chatbot, ChatbotController üzerinden ChatbotService'e mesaj gönderir. Gemini anahtarı bulunduğunda modelden spor salonu bağlamına uygun cevap istenir. Bağlantı kurulamazsa sistem tamamen devre dışı kalmak yerine yerel kurallarla temel cevap üretmeye devam eder. Bu yaklaşım hata toleransını artırır.

5. KURULUM VE ÇALIŞTIRMA

5.1 Gerekli Yazılımlar

- Windows 10/11 veya .NET 9 destekleyen bir işletim sistemi.
- .NET 9 SDK.
- Visual Studio 2022 veya Visual Studio Code.
- SQL Server ve tercihen SQL Server Management Studio.
- İnternet bağlantısı (NuGet paketlerinin ilk kurulumu ve Gemini kullanımı için).

5.2 Projenin Hazırlanması

1. ZIP dosyasını bir klasöre çıkarın.
2. SmartGymSystem.sln dosyasını Visual Studio ile açın.
3. appsettings.json içindeki DefaultConnection değerini kendi SQL Server kurulumunuza göre düzenleyin.
4. Terminalde dotnet restore komutuyla NuGet paketlerini yükleyin.
5. dotnet build komutuyla projeyi derleyin.
6. dotnet run komutunu çalıştırın veya Visual Studio'da HTTPS profiliyle başlatın.

```
dotnet restore
dotnet build
dotnet run
```

5.3 Veritabanının Oluşturulması

Uygulama başlatılırken Program.cs içinde DbInitializer.Initialize çağrılır. DbInitializer, Database.EnsureCreated ile veritabanı ve tabloları oluşturur; paketler, kategoriler, ürünler, egzersizler ve örnek kullanıcı kayıtları yoksa başlangıç verilerini ekler. Bu nedenle ilk çalıştırmada ayrıca migration komutu gerekmemektedir.

Güvenlik notu

Projeyi gerçek ortamda yayımlamadan önce bağlantı dizesi, JWT anahtarı ve Gemini API anahtarı kaynak kod dışında User Secrets, ortam değişkeni veya güvenli bir secret store içinde tutulmalıdır. Başlangıç kullanıcılarının parolaları da değiştirilmelidir.

5.4 Yapılandırma Örneği

```
{
  "ConnectionStrings": {
    "DefaultConnection":
    "Server=...;Database=SmartGymDb;Trusted_Connection=True;TrustServerCertificate=True"
  },
  "Jwt": {
    "Key": "güçlü-ve-uzun-bir-gizli-anahtar",
    "Issuer": "SmartGymSystem",
    "Audience": "SmartGymSystemUsers"
  },
  "Gemini": {
    "ApiKey": "ortam-değişkeninden-okunmalıdır"
  }
}
```

6. UYGULAMA EKRLANLARI

Bu bölümde uygulamanın çalışan sürümünden alınan ekran görüntüleri ve ilgili backend işlevleri açıklanmıştır. Her görsel, kaynak kodda bulunan controller, model ve view bileşenleriyle ilişkilendirilmiştir.

6.1 Ana Sayfa



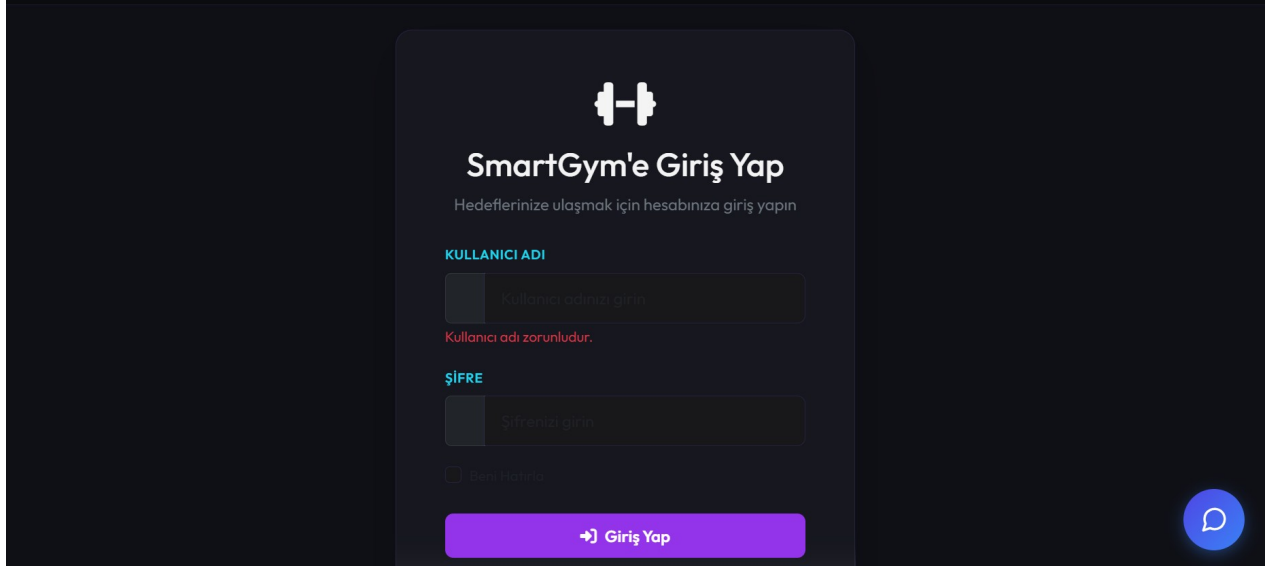
Şekil 3. Ana Sayfa

Uygulamanın karşılama ekranı; giriş, kayıt, spor mağazası ve yapay zekâ destekli asistan erişimlerini tek noktada sunmaktadır.

Backend bağlantısı

HomeController.Index ve Views/Home/Index.cshtml tarafından sunulur. Ana navigasyon, rol durumuna göre panel bağlantılarını dinamik olarak gösterir.

6.2 Giriş ve Form Doğrulama

The image shows a login form for 'SmartGym'e Giriş Yap'. The form is centered on a dark background. At the top, there is a dumbbell icon. Below it, the title 'SmartGym'e Giriş Yap' is displayed in white, followed by the subtitle 'Hedeflerinize ulaşmak için hesabınıza giriş yapın'. The form contains two input fields: 'KULLANICI ADI' (Username) and 'ŞİFRE' (Password). The 'KULLANICI ADI' field has a red error message 'Kullanıcı adı zorunludur.' (Username is required.) below it. The 'ŞİFRE' field has a red error message 'Şifre zorunludur.' (Password is required.) below it. There is a checkbox labeled 'Ben Hatırlıyorum' (Remember me) below the password field. At the bottom of the form is a blue button with a right arrow icon and the text 'Giriş Yap' (Login). In the bottom right corner of the screen, there is a blue circular icon with a white speech bubble.

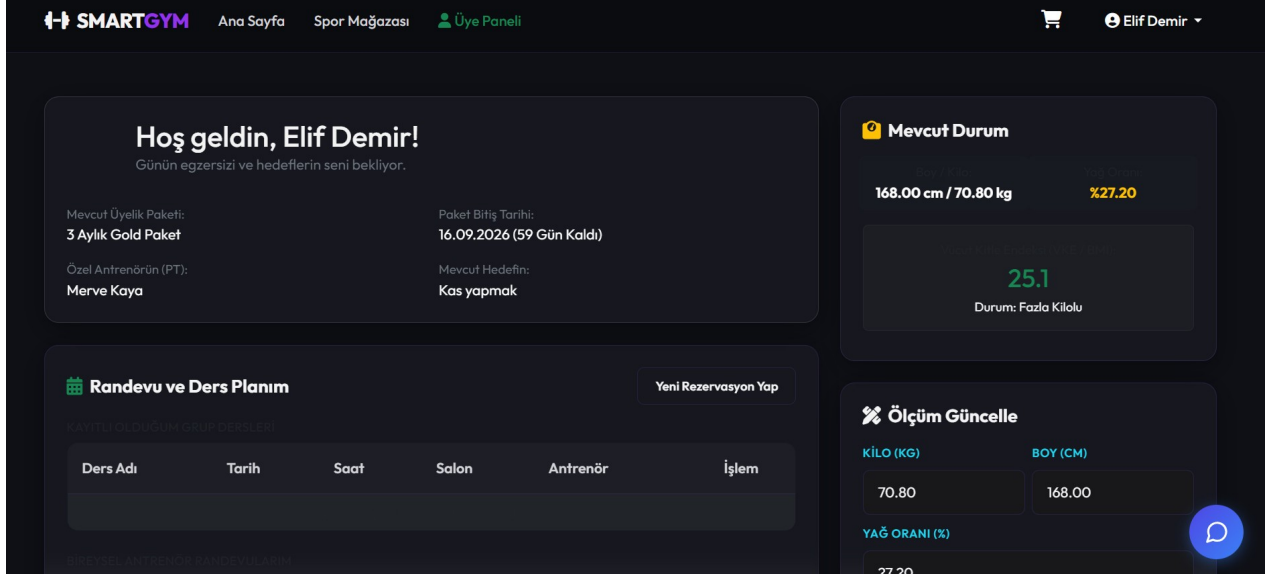
Şekil 4. Giriş ve Form Doğrulama

Kullanıcı adı ve şifre ile oturum açma ekranıdır. Zorunlu alan kontrolleri ModelState ve istemci tarafı doğrulama bileşenleriyle kullanıcıya gösterilmektedir.

Backend bağlantısı

AccountController.Login POST eylemi ModelState kontrolü, şifre doğrulaması, claim üretimi ve Cookie SignInAsync işlemlerini yürütür.

6.3 Üye Kontrol Paneli



Şekil 5. Üye Kontrol Paneli

Üyelik paketi, bitiş tarihi, atanmış antrenör, hedef, güncel vücut ölçümleri ve randevu planı tek ekranda sunulmaktadır.

Backend bağlantısı

MemberController.Dashboard; Member, User, MembershipPlan, AssignedTrainer, ProgressRecords ve randevu verilerini ilişkileriyle birlikte yükler.

6.4 Akıllı Antrenman Programı

[← Programlarıma Dön](#)

Akıllı Eşleştirme: Kas yapmak Antrenmanı

Hazırlayan: Akıllı Sistem | Hedef: Kas yapmak

Görsel	Egzersiz	Kategori	Gün	Set x Tekrar	Dinlenme	Açıklama / Notlar
	Bench Press	Göğüs	Pazartesi	4 x 10	90 Sn.	Son set yardımcı max kilo.
	Tricep Pushdown	Kol	Pazartesi	3 x 12	60 Sn.	Dirsekler sabit kalsın.
	Deadlift	Sırt	Çarşamba	4 x 8	120 Sn.	Bel formuna dikkat edin.
	Barfiks (Pull-up)	Sırt	Çarşamba	3 x 8	90 Sn.	Negatif yavaş yapın.
	Squat (Ağırlıklı)	Bacak	Cuma	4 x 10	120 Sn.	Derin çömelme.

Şekil 6. Akıllı Antrenman Programı

Hedefe göre oluşturulan program; egzersiz, kategori, gün, set-tekrar, dinlenme ve açıklama bilgileriyle ayrıntılı biçimde listelenmektedir.

Backend bağlantısı

WorkoutDetails eylemi, WorkoutProgramDetail kayıtlarını Exercise ilişkisiyle birlikte getirir ve programı tablo halinde gösterir.

6.5 Akıllı Program Sihirbazı



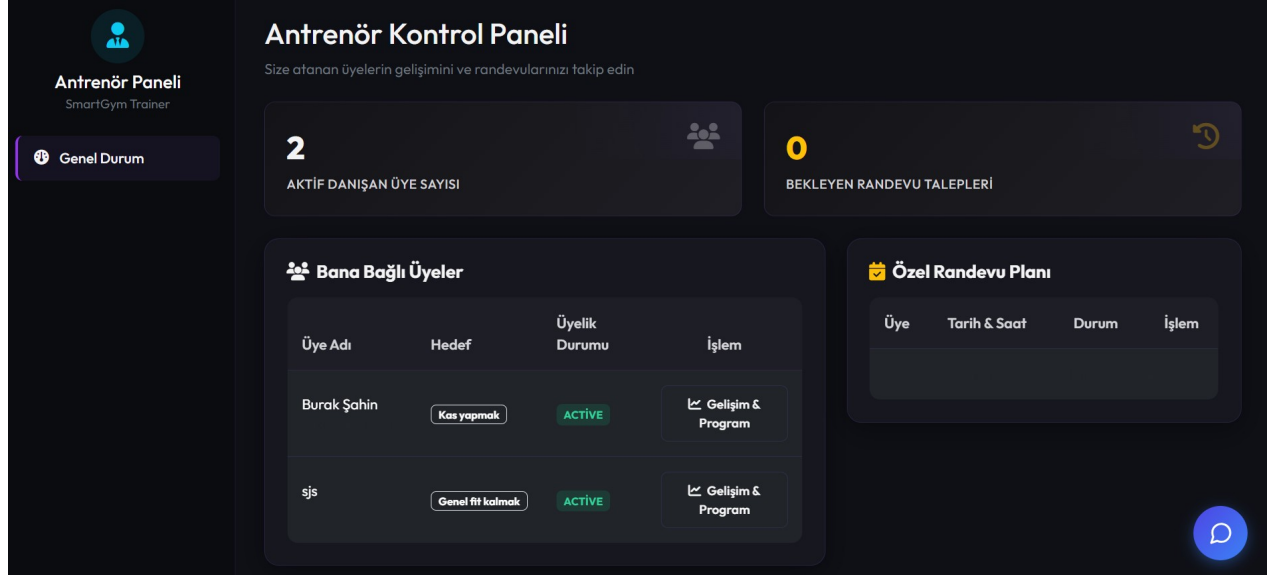
Şekil 7. Akıllı Program Sihirbazı

Kilo verme, kas yapma, kondisyon artırma ve genel fit kalma hedeflerinden biri seçilerek otomatik program üretilmektedir.

Backend bağlantısı

ApplyRecommendation hedef parametresini alır, yeni WorkoutProgram kaydı açar ve hedefe uygun egzersiz detaylarını otomatik ekler.

6.6 Antrenör Kontrol Paneli



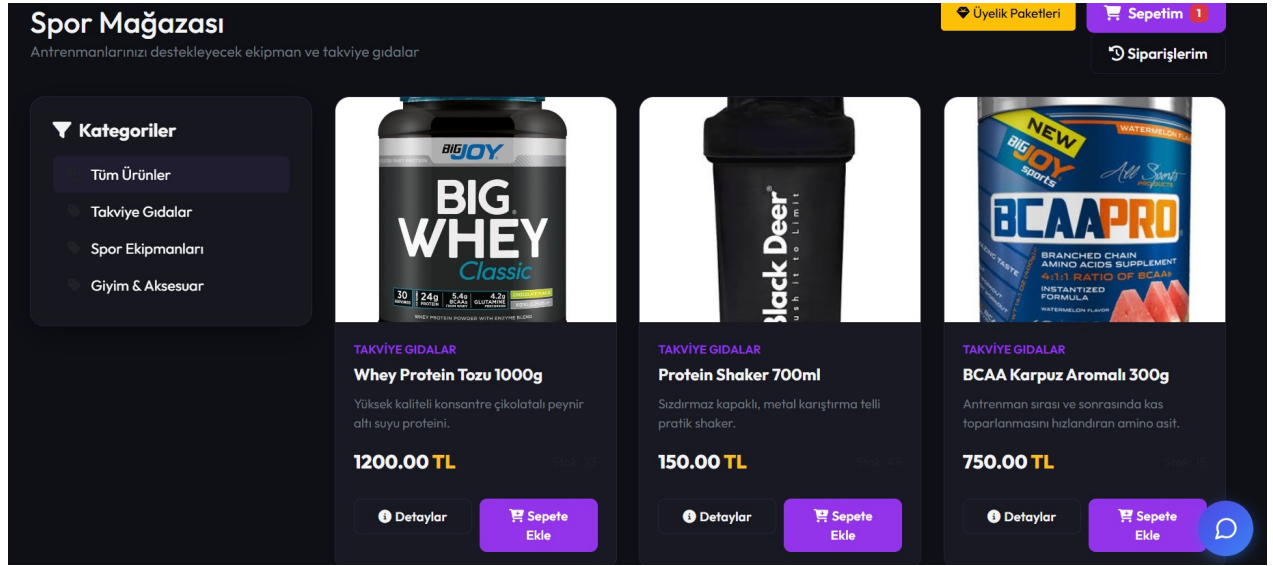
Şekil 8. Antrenör Kontrol Paneli

Antrenör; kendisine atanan üyeleri, üyelik durumlarını, hedefleri ve özel randevu taleplerini görüntüleyebilmektedir.

Backend bağlantısı

TrainerController.Dashboard, yalnızca giriş yapan antrenöre atanmış üyeleri ve yaklaşan randevuları filtreler.

6.7 Spor Mağazası



Şekil 9. Spor Mağazası

Kategori bazlı ürün listeleme, ürün detayı, stok bilgisi, sepet ve sipariş süreçlerini içeren e-ticaret modülüdür.

Backend bağlantısı

ShopController.Index ürünleri kategoriyle birlikte getirir; seçilen categoryId varsa sorguya filtre uygular ve üyenin sepet sayısını hesaplar.

6.8 Üyelik Paketleri

[← Kontrol Paneline Dön](#)

Üyelik & Abonelik Paketleri

Size en uygun salon aboneliğini seçerek hedeflerinize hemen başlayın

 1 Aylık Standart Paket 1 aylık standart salon ve kardiyo alanı kullanımı. Süre: 1 Ay Fiyat: 300.00 TL Satin Al	Aktif Paketiniz  3 Aylık Gold Paket 3 aylık salon kullanımı ve 2 adet ücretsiz bireysel antrenör seansı. Süre: 3 Ay Fiyat: 800.00 TL Kullanımda	 6 Aylık Premium Paket 6 aylık sınırsız salon, grup dersleri ve SPA alanı kullanımı. Süre: 6 Ay Fiyat: 1500.00 TL Satin Al	 12 Aylık Platinum Paket 12 aylık her şey dahil üyelik, sınırsız grup dersi ve özel dolap hakkı. Süre: 12 Ay Fiyat: 2600.00 TL Satin Al
--	---	---	--



Şekil 10. Üyelik Paketleri

Farklı süre ve fiyat seçeneklerine sahip üyelik paketleri karşılaştırılmakta; aktif paket ayrıca işaretlenmektedir.

Backend bağlantısı

ShopController.Plans aktif paketleri listeler; BuyPlan üyelik başlangıç/bitiş tarihlerini günceller ve Payment kaydı oluşturur.

6.9 Yönetici - Üye Yönetimi

Üye Bilgisi	Üyelik Paketi	Bitiş Tarihi	Antrenör	Hedef	Durum	İşlemler
Elif Demir	3 Aylık Gold Paket	16.09.2026	Merve Kaya	Kas yapmak	ACTIVE	[Edit] [Delete]
Burak Şahin	6 Aylık Premium Paket	25.12.2026	Ahmet Yılmaz	Kas yapmak	ACTIVE	[Edit] [Delete]
berkay eldut	12 Aylık Platinum Paket	09.07.2027	Atanmadı	Genel fit kalmak	ACTIVE	[Edit] [Delete]
sjs	1 Aylık Standart Paket	09.08.2026	Ahmet Yılmaz	Genel fit kalmak	ACTIVE	[Edit] [Delete]

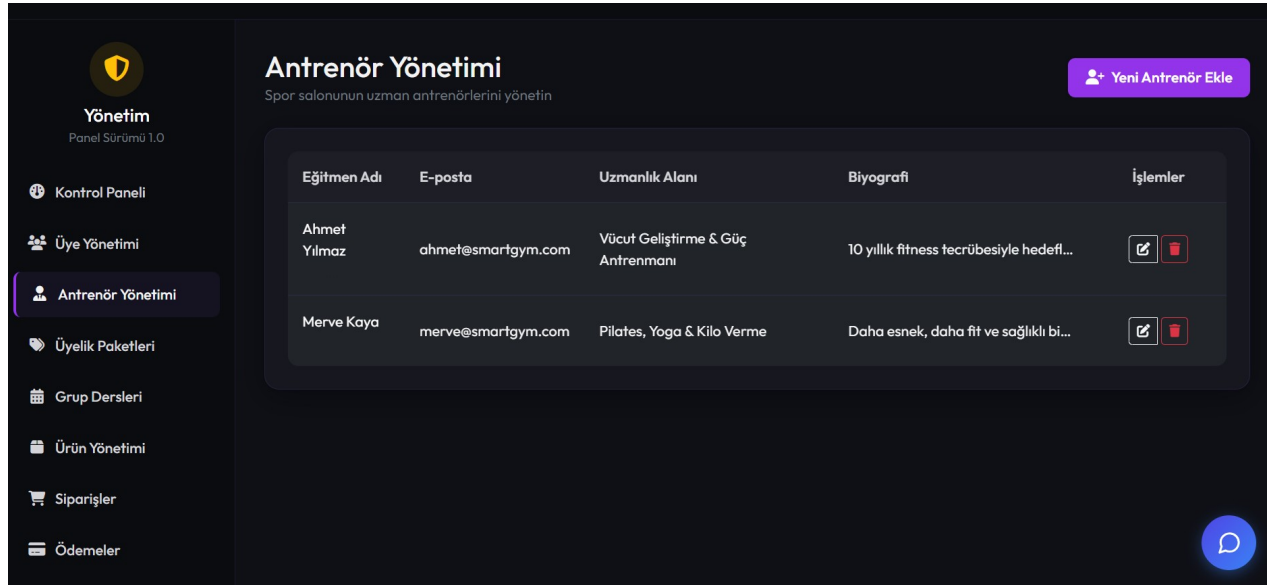
Şekil 11. Yönetici - Üye Yönetimi

Yönetici; üyeleri, paketleri, bitiş tarihlerini, atanmış antrenörleri, hedefleri ve durum bilgilerini yönetebilmektedir.

Backend bağlantısı

AdminController.Members gerekli ilişkileri Include ile yükler. Düzenleme ve silme eylemleri yalnızca Admin rolüne açıktır.

6.10 Yönetici - Antrenör Yönetimi



Şekil 12. Yönetici - Antrenör Yönetimi

Antrenör hesapları, uzmanlık alanları ve biyografi bilgileri yönetim paneli üzerinden eklenebilmekte ve güncellenebilmektedir.

Backend bağlantısı

AdminController.Trainers kullanıcı hesabı ile antrenör profilini birlikte yönetir; uzmanlık ve biyografi alanlarını saklar.

6.11 Yönetici - Üyelik Paketi Yönetimi

Paket Adı	Açıklama	Süre (Ay)	Fiyat	Durum	İşlemler
1 Aylık Standart Paket	1 aylık standart salon ve kardiyo alanı kullanımı.	1 Ay	£300.00	Aktif	
3 Aylık Gold Paket	3 aylık salon kullanımı ve 2 adet ücretsiz bireysel antrenör seansı.	3 Ay	£800.00	Aktif	
6 Aylık Premium Paket	6 aylık sınırsız salon, grup dersleri ve SPA alanı kullanımı.	6 Ay	£1,500.00	Aktif	
12 Aylık Platinum Paket	12 aylık her şey dahil üyelik, sınırsız grup dersi ve özel dolap hakkı.	12 Ay	£2,600.00	Aktif	

Şekil 13. Yönetici - Üyelik Paketi Yönetimi

Paket adı, açıklama, süre, fiyat ve aktiflik durumu CRUD işlemleriyle yönetilmektedir.

Backend bağlantısı

MembershipPlan kayıtları süre, fiyat ve aktiflik bilgileriyle CreatePlan, EditPlan ve DeletePlan eylemlerinden yönetilir.

6.12 Yönetici - Grup Dersleri

SMARTGYM Ana Sayfa Spor Mağazası **Yönetici Paneli** Sistem Yöneticisi

Yönetim
Panel Sürümü 1.0

Kontrol Paneli
Üye Yönetimi
Antrenör Yönetimi
Üyelik Paketleri
Grup Dersleri
Ürün Yönetimi
Siparişler

Grup Dersleri Yönetimi

Programlanmış grup egzersiz derslerini planlayın ve katılımcıları takip edin

[+ Yeni Ders Programla](#)

Ders Adı	Tarih	Saat	Salon	Eğitmen	Kapasite Durumu	İşlemler
deneme	08.07.2026	15:40 - 16:40	deneme	Merve Kaya	1 / 10	Düzenle Sil
Zumba Blast	05.07.2026	18:30 - 19:30	Grup Ders Salonu A	Merve Kaya	0 / 25	Düzenle Sil
Crossfit Giriş	04.07.2026	19:00 - 20:00	Crossfit Alanı	Ahmet Yilmaz	1 / 12	Düzenle Sil
Power Pilates	04.07.2026	10:00 - 11:00	Aletli Stüdyo	Merve Kaya	0 / 15	Düzenle Sil

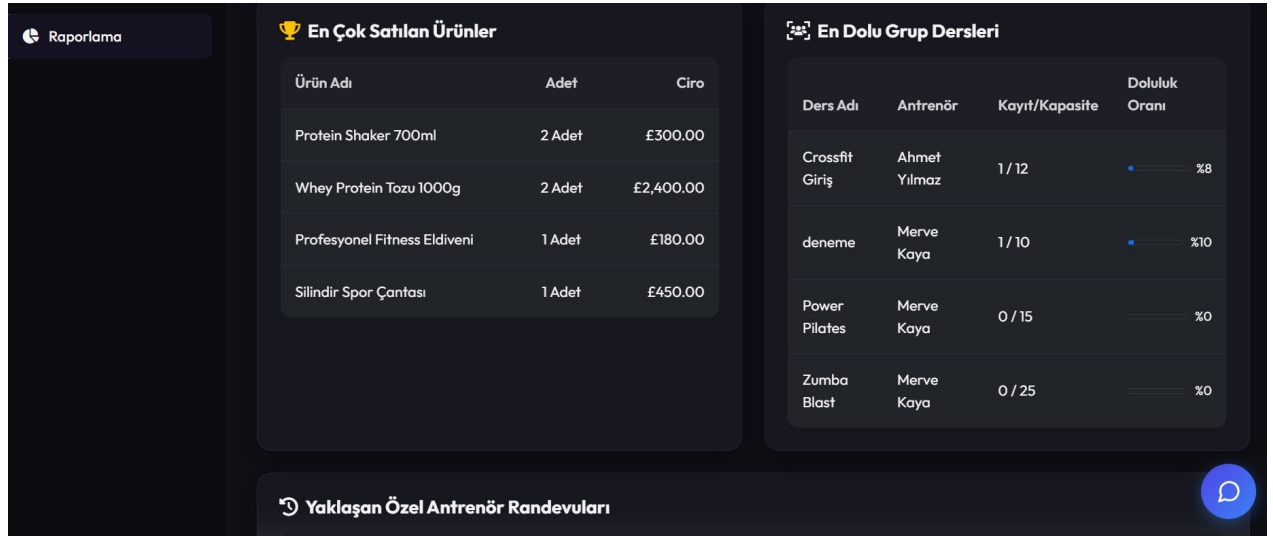
Şekil 14. Yönetici - Grup Dersleri

Ders tarihi, saati, salonu, eğitmeni ve kapasite dolulukları merkezi olarak planlanmaktadır.

Backend bağlantısı

GroupClass kayıtları eğitmen, tarih, saat, salon ve kapasite bilgilerini içerir; kayıt sayısı üzerinden doluluk bilgisi hesaplanır.

6.13 Raporlama Paneli



Şekil 15. Raporlama Paneli

En çok satılan ürünler, ciro ve en dolu grup dersleri gibi yönetsel göstergeler raporlama ekranında birleştirilmektedir.

Backend bağlantısı

AdminController.Reports toplam gelir, üyelik durumları, düşük stok, yaklaşan randevular, popüler dersler ve en çok satılan ürünleri toplar.

7. TEST, GÜVENLİK VE PERFORMANS

7.1 Uygulanan Kontroller

- Formlarda DataAnnotations ve ModelState doğrulaması.
- POST işlemlerinin büyük bölümünde ValidateAntiForgeryToken kullanımı.
- Admin, Trainer ve Member ekranlarında rol bazlı Authorize kontrolü.
- Randevu ve program sorgularında profil sahipliği kontrolü.
- Sepete ekleme ve checkout aşamalarında stok doğrulaması.
- Düşük stok, üyelik ve sipariş durumları için yönetim göstergeleri.
- Serilog ile bilgi, uyarı ve hata kayıtları.
- MemoryCache ile tekrar eden API sorgularının azaltılması.

7.2 Örnek Test Senaryoları

Senaryo	Beklenen Sonuç	Durum
Boş kullanıcı adıyla giriş	Zorunlu alan hatası gösterilir.	Başarılı
Member rolüyle /Admin erişimi	Erişim reddedilir.	Başarılı
Stoktan fazla ürün ekleme	İşlem durdurulur ve stok uyarısı gösterilir.	Başarılı
Aynı grup dersine tekrar katılma	Mükerrer aktif kayıt engellenir.	Başarılı
Antrenörün bağlı olmayan üyeye erişmesi	NotFound/Yetkisiz sonuç döner.	Başarılı
API üye listesi ikinci çağrı	Veri önbellekten döndürülür.	Başarılı

7.3 Güvenlik ve Kalite İyileştirmeleri

Proje eğitim amaçlı olarak çalışır durumdadır; ancak üretim ortamı için aşağıdaki geliştirmeler önerilir:

- Parola saklama işleminde düz SHA-256 yerine salt ve uyarlanabilir maliyet kullanan ASP.NET Core PasswordHasher, Argon2 veya bcrypt tercih edilmelidir.
- Program.cs ve TokenService içindeki varsayılan JWT anahtarı kaldırılmalı; anahtar yalnızca güvenli yapılandırmadan gelmelidir.

- ApiMembersController üzerine JWT tabanlı Authorize eklenmeli ve CreateMember eylemi uygun role sınırlandırılmalıdır.
- Sepet güncelleme ve silme işlemlerinde CartItem kaydının giriş yapan üyeye ait olduğu ayrıca doğrulanmalıdır.
- Ödeme işlemi gerçek bir ödeme sağlayıcısıyla entegre edilecekse webhook doğrulaması, idempotency ve transaction yönetimi eklenmelidir.
- Database.EnsureCreated yerine migration tabanlı sürümleme kullanılmalıdır.
- Controller sınıflarındaki kapsamlı iş kuralları uygulama servislerine taşınarak test edilebilirlik artırılmalıdır.
- Birim testleri, entegrasyon testleri ve API dokümantasyonu için xUnit ve Swagger/OpenAPI eklenmelidir.

Değerlendirme

Bu maddeler projenin başarısız olduğu anlamına gelmez. Mevcut sürüm, backend eğitiminde beklenen MVC, veritabanı, yetkilendirme, API, raporlama ve tasarım deseni konularını aynı projede göstermektedir; öneriler üretim kalitesine geçiş içindir.

8. SONUÇ VE GELİŞTİRME ÖNERİLERİ

SmartGymSystem projesinde spor salonunun temel operasyonları tek bir web uygulamasında başarıyla birleştirilmiştir. Yönetici, antrenör ve üye rolleri için ayrı iş akışları geliştirilmiş; rol tabanlı erişimle kullanıcıların yalnızca yetkili oldukları işlemleri yapması sağlanmıştır. Üyelik yönetimi, randevu, grup dersi, vücut ölçümü, antrenman programı, mağaza, stok, sipariş, ödeme ve raporlama modülleri ilişkisel veritabanı üzerinde bütünleşik biçimde çalışmaktadır.

Backend tarafında hem Entity Framework Core hem Dapper kullanılması, Repository ve Unit of Work desenlerinin uygulanması, Cookie ve JWT kimlik doğrulamanın aynı projede bulunması, MemoryCache ve Serilog entegrasyonu ile Excel/PDF çıktıların üretilmesi projenin teknik kapsamını güçlendirmiştir. Akıllı program sihirbazı ve chatbot ise geleneksel yönetim sistemi işlevlerini kullanıcı destek özellikleriyle tamamlamıştır.

8.1 Gelecekte Eklenebilecek Özellikler

- Gerçek ödeme sistemi ve e-fatura entegrasyonu.
- E-posta/SMS randevu ve üyelik bitiş bildirimleri.
- Antrenman ilerlemesi için grafikler ve hedef karşılaştırmaları.
- Mobil uygulama veya Progressive Web App desteği.
- Swagger/OpenAPI dokümantasyonu ve versiyonlanmış API.
- Docker ile uygulama ve SQL Server kurulumu.
- Bulut dağıtımı, merkezi loglama ve sağlık kontrolü uç noktaları.
- Daha gelişmiş, veri tabanlı antrenman öneri motoru.

9. KAYNAKÇA

- [1] Microsoft Learn, “ASP.NET Core MVC Overview”, <https://learn.microsoft.com/aspnet/core/mvc/>
- [2] Microsoft Learn, “Entity Framework Core Documentation”, <https://learn.microsoft.com/ef/core/>
- [3] Microsoft Learn, “SQL Server Documentation”, <https://learn.microsoft.com/sql/sql-server/>
- [4] Dapper, NuGet Package Documentation, <https://www.nuget.org/packages/Dapper>
- [5] Serilog Documentation, <https://serilog.net/>
- [6] ClosedXML Documentation, <https://docs.closedxml.io/>
- [7] QuestPDF Documentation, <https://www.questpdf.com/documentation/getting-started.html>
- [8] Google AI for Developers, “Gemini API Documentation”, <https://ai.google.dev/gemini-api/docs>
- [9] Bootstrap Documentation, <https://getbootstrap.com/docs/5.3/>
- [10] SmartGymSystem proje kaynak kodları ve uygulama ekran görüntüleri, Berkay Eldut, 2026.