



4. Dönem / 1. Grup

**BACKEND
BİTİRME PROJESİ RAPORU**

AdeSea Yacht Manufacturing & Sales

Lüks Yat Tasarım, Satış ve Üretim Simülasyon Platformu

Adem Eraslan

Sena Üstün

Temmuz, 2026

softITo Yazılım Bilişim Akademisi, İstanbul

İçindekiler

1. GİRİŞ	3
1.1. Projenin Amacı.....	3
1.2. Proje Kapsamı ve Kısıtları	3
2. MATERYAL VE YÖNTEM.....	4
2.1. Kullanılan Teknolojiler	4
2.1.1. API (Backend) Katmanı.....	4
2.1.2. MVC (Frontend) Katmanı:	4
2.2. Veritabanı Mimarisi	4
3. SİSTEM ÖZELLİKLERİ VE SAYFA HARİTASI.....	6
3.1. Müşteri Sayfaları	6
3.2. Yönetim Paneli	7
4. KURULUM VE ÇALIŞTIRMA	8
5. GELİŞTİRME SÜRECİ VE EKİP ÇALIŞMASI.....	9
6. BENİM KATKILARIM (Adem Eraslan).....	10
7. SONUÇ	11
8. KAYNAKÇA.....	12

1. GİRİŞ

1.1. Projenin Amacı

AdeSea projesinin temel amacı, geleneksel lüks yat üretim sürecini dijital bir platforma taşımaktır. Proje, softITo programı kapsamında bir bitirme ödevi olarak hazırlanmıştır. Bu proje ile, normal şartlarda yüz yüze yürütülen özel yat sipariş süreçleri tek bir ekranda birleştirilmiştir.

Proje sayesinde müşterilere sunulan yenilikler şunlardır:

- Müşteriler sisteme giriş yaparak yatın gövde malzemesini, iç mekanını ve motorunu seçebilmektedir.
- Ekstra donanımlar eklenerek yat sıfırdan konfigüre edilebilmektedir.
- Sipariş verildikten sonra çalışan üretim simülasyonu sayesinde, yatın üretim aşamaları izlenebilmektedir.

1.2. Proje Kapsamı ve Kısıtları

Uygulama, müşteriler ve yöneticiler için iki farklı arayüz içerecek şekilde kurgulanmıştır. Mikro-Monolit bir mimari tercih edilmiştir. Sistemin kapsadığı temel modüller şu şekildedir:

Müşteri Modülü: Kullanıcı üyeliği, interaktif yat tasarım sihirbazı ve gerçek zamanlı üretim takibi.

Yönetici Modülü: Stok takibi, sipariş onay süreçleri, içerik moderasyonu ve iş zekası raporlaması.

Eğitim odaklı bir proje olması nedeniyle belirlenen kısıtlar şunlardır:

- Üretim süreci gerçek bir fabrikaya bağlı değildir. Aşamalar yazılımsal bir simülasyon motoru tarafından ilerletilmektedir.
- Satış asistanı olarak Google Gemini API entegre edilmiştir. Ancak bu asistan sadece yatçılık alanında cevap verecek şekilde kısıtlandırılmıştır.
- Geliştirme ve test süreçleri sunucu ortamı yerine yerel veritabanı (LocalDB) üzerinde gerçekleştirilmiştir.
- Projede yerelleştirme yapılarak Türkçe, İngilizce ve Arapça dil seçenekleri sisteme dahil edilmiştir.

2. MATERYAL VE YÖNTEM

Geliştirme sürecinde karmaşıklığı önlemek amacıyla kod tabanı API ve MVC olarak ikiye ayrılmıştır. Veri işlemleri ve iş kuralları API tarafında tasarlanırken, MVC katmanı sadece API ile haberleşerek verileri ekrana yansıtmıştır.

2.1. Kullanılan Teknolojiler

Projenin geliştirilmesinde modern web teknolojilerinden faydalanılmıştır.

2.1.1. API (Backend) Katmanı

- **Çerçeve (Framework):** Proje ASP.NET Core 10.0 Web API mimarisi ile oluşturulmuştur.
- **Veri Erişimi:** Veritabanı kayıt işlemlerinde güvenlik amacıyla Entity Framework Core kullanılmıştır. Yönetim panelindeki raporlamalar ve veri filtrelemeleri gibi performans gerektiren noktalarda ise Dapper tercih edilmiştir.
- **Güvenlik:** Kimlik doğrulama süreçleri için Identity ve JWT (JSON Web Token) teknolojilerinden yararlanılmıştır.
- **Arka Plan İşlemleri:** Sipariş simülasyonunu yönetmek için arka planda sürekli çalışan bir Background Service (IHostedService) geliştirilmiştir.
- **Gerçek Zamanlı İletişim:** Üretim durumunu arayüzde anlık güncellemek için SignalR kütüphanesi entegre edilmiştir.
- **Yardımcı Paketler:** Sistem hatalarının kaydı için Serilog, sözleşme oluşturmak için QuestPDF ve Excel çıktıları için ClosedXML kullanılmıştır.

2.1.2. MVC (Frontend) Katmanı:

- **Çerçeve (Framework):** Kullanıcı arayüzü ASP.NET Core 10.0 MVC ile geliştirilmiştir. API tarafına yapılan istekler HttpClientFactory üzerinden yönetilmiştir.
- **Arayüz Tasarımı:** HTML5, CSS3, Bootstrap 5 ve Vanilla JavaScript kullanılarak modern bir görünüm elde edilmiştir.
- **Etkileşim:** Yat tasarım sayfasındaki form etkileşimleri JavaScript ile sağlanmıştır.
- **Görselleştirme:** Yöneticiler için hazırlanan iş zekası grafikleri Chart.js ile çizilmiştir.
- **QR Kod:** Sipariş tamamlandığında oluşan QR kodu, harici bir servise bağlanmadan projeye dahil edilen QRCoder kütüphanesi ile üretilmiştir.

2.2. Veritabanı Mimarisi

Veritabanı MS SQL Server üzerinde yapılandırılmış ve toplamda 22 tablo kullanılmıştır. Veri tekrarını önlemek ve sorgu hızını artırmak için tablolar gruplara ayrılmıştır:

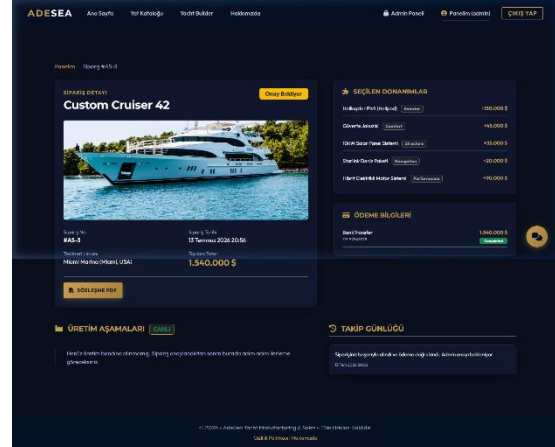
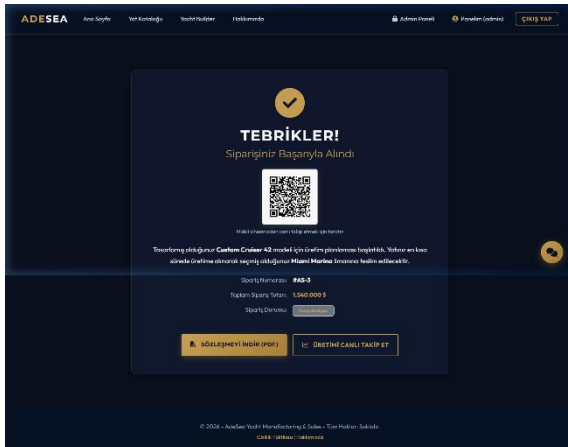
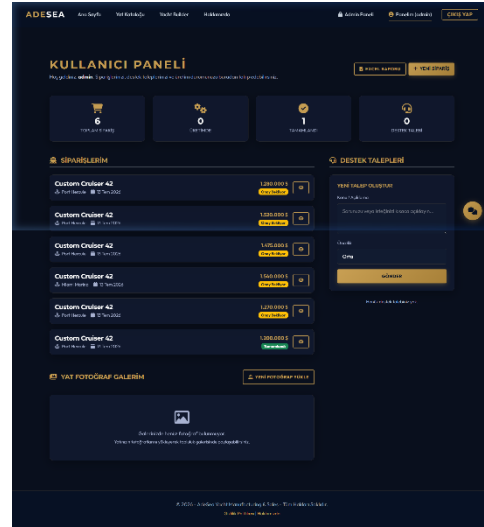
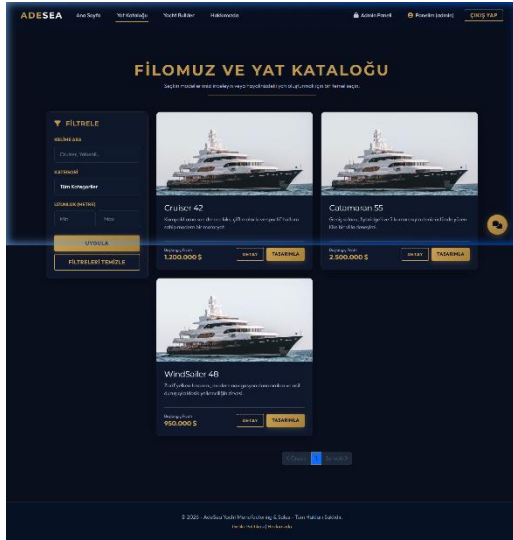
- **Kullanıcı Modülü:** Kullanıcı bilgileri, yetkiler ve bildirimler hesap tablolarında tutulmuştur.
- **Katalog Modülü:** Yat modelleri, motor seçenekleri ve donanımlar sınıflandırılarak saklanmıştır.
- **Finans Modülü:** Gelen siparişler, ödeme işlemleri ve taksit planları kayıt altına alınmıştır.
- **Üretim Modülü:** Depodaki malzemeler ve siparişlerin simülasyon aşamaları izlenmiştir.

- **İçerik Modülü:** Müşteri yorumları, dinamik site içerikleri ve müşteri destek talepleri için ayrı tablolar oluşturulmuştur.

3. SİSTEM ÖZELLİKLERİ VE SAYFA HARİTASI

3.1. Müşteri Sayfaları

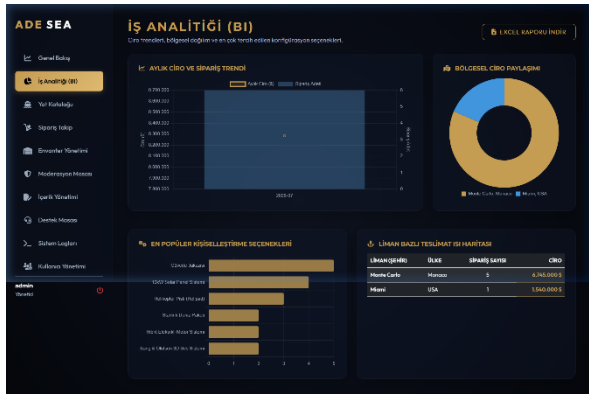
- Müşterilerin sistemi rahat kullanabilmesi için sayfalar kullanıcı dostu olarak tasarlanmıştır. Öne çıkan müşteri sayfaları şunlardır:
- Ana Sayfa ve Katalog:** Ziyaretçileri yat görselleri ve yapay zeka asistanı karşılamaktadır. Katalog bölümünde bütçe veya istenilen özelliklere göre hızlı filtreleme yapılabilmektedir.
- Yat Tasarım Sihirbazı:** Uygulamanın merkezinde yer alan bu ekranda, müşteriler gövde renginden ekstra donanımlara kadar detaylı seçimler yapabilmektedir. Seçimlere bağlı olarak yat fiyatı anlık şekilde hesaplanmaktadır.
- Rastgele Tasarım:** Hızlı seçim yapmak isteyen kullanıcılar için "Zarları At" butonu eklenmiştir. Bu özellik sayesinde birbiriyle uyumlu parçalar birleştirilerek otomatik bir yat konfigürasyonu sunulmaktadır.
- Sipariş ve Ödeme İşlemleri:** Tasarımı tamamlanan yatların sipariş ekranında teslimat limanı seçimi yapılmaktadır. Ödeme aşamasında ise kullanıcılara peşin ödeme veya taksitli ödeme planı gibi esnek seçenekler sunulmaktadır.
- Kullanıcı Paneli ve Üretim Takibi:** Sipariş onaylandıktan sonra müşteriler, SignalR altyapısıyla yatlarının üretim aşamalarını sayfa yenilemeye gerek kalmadan izleyebilmektedir. Ayrıca bu panel üzerinden kendi yatlarına ait fotoğrafları "Fotoğraf Galerim" modülüne yükleyebilmekte ve yetkililerle iletişime geçmek için destek talebi (ticket) oluşturabilmektedir.



3.2. Yönetim Paneli

Sistemin kontrol edilmesi ve verilerin analiz edilmesi amacıyla yönetim paneli geliştirilmiştir. Yönetim panelinin özellikleri şunlardır:

- **İş Zekası (BI) Raporları:** En çok satılan yat modelleri ve tercih edilen renkler grafiksel olarak görselleştirilmiştir. Bu raporlar Excel veya PDF formatında indirilebilmektedir.
- **Katalog ve Şablon Yönetimi:** Yöneticiler sisteme yeni yat kategorileri, şablonlar ve ekstra donanım seçenekleri (YachtParts) ekleyebilmekte; mevcut donanımların fiyatlarını güncelleyebilmektedir.
- **Envanter ve Stok Yönetimi:** Depodaki malzemelerin stok durumu izlenebilmekte ve stok seviyeleri kritik sınırın altına düştüğünde sistem otomatik uyarı vermektedir.
- **İçerik Moderasyonu ve Müşteri Destek:** Müşterilerden gelen yat fotoğrafları ve yorumlar moderasyon onayından geçtikten sonra yayına alınmaktadır. Aynı zamanda müşterilerin açtığı destek biletleri (ticket) bu panelden yanıtlanıp kapatılabilmektedir.
- **Sistem Logları ve Dinamik Sayfalar:** "Hakkımızda" gibi sayfaların içerikleri kodlama gerektirmeden panel üzerinden (CMS yapısıyla) düzenlenebilmektedir. Ayrıca Serilog tarafından veritabanına yazılan sistem hataları (Error/Info) özel bir log izleme ekranı üzerinden filtrelenerek incelenebilmektedir.



The CMS dashboard allows managing content. It features a table with columns: 'SAYFA ADI (PAGE NAME)', 'SAYFA BAĞLI (PAGE LINK)', 'SON GÜNCELLEME (LAST UPDATE)', and 'İŞLEMLER (ACTIONS)'. The table lists pages like 'About Us', 'Hakkımızda', 'Yat Katalogu', and 'Privacy'. The 'İŞLEMLER' column has buttons for 'DÜZENLE' (Edit) and 'YENİLE' (Refresh).

The system log viewer displays a table of logs with columns: 'TARİH / SAAT (DATE / TIME)', 'SEVİYE (LEVEL)', 'LOG MESAJI (LOG MESSAGE)', 'HATA DETAYI (DESCRIPTION)', and 'ALINMIŞ (RECEIVED)'. The table lists logs from 2023-07-01 to 2023-07-03, including messages like 'Production Simulation Modelled Service is starting' and 'ADESEA API is starting up'.

The inventory and warehouse management dashboard shows a table of items with columns: 'MALZEME ADI (MATERIAL NAME)', 'MEVCUT STOK (CURRENT STOCK)', 'BİRİM (UNIT)', 'ALANIN EĞİLİ (AREA TENDENCY)', 'STOK DURUMU (STOCK STATUS)', 'DURUM (STATUS)', and 'İŞLEMLER (ACTIONS)'. The table lists items like 'Ağır Beton Lanta', 'Çelik Profil', 'Fiberglas Boyut', 'Hafif Çelik İskelet', 'Aksel Bağlayıcı Kiti', 'Mara Kablo Seti', 'Saklar Panel Çiftli', 'Starlink Dışarı Çıkarıcı', 'Tutak Alınan Kablo', 'Yüksek Pnömatik Çiftli', and 'Yüksek Pnömatik Çiftli'.

4. KURULUM VE ÇALIŞTIRMA

Projeyi çalıştırmak için izlenmesi gereken adımlar yapılandırılmıştır. Ön gereksinim olarak sistemde .NET 10 SDK ve SQL Server LocalDB kurulu olması gerekmektedir.

- **Veritabanı Hazırlığı:** Öncelikle LocalDB başlatılır. Ardından veritabanı tablolarını oluşturmak için komut satırından migration işlemi uygulanır.
- **Başlangıç Verileri:** Tablolar oluştuktan sonra, sisteme örnek yatları ve malzemeleri eklemek amacıyla proje dizinindeki `seed.sql` dosyası veritabanında çalıştırılır.
- **Uygulamanın Başlatılması:** Projeyi tek bir işlemle başlatmak için `.slnlaunch` dosyası hazırlanmıştır. Bu dosya sayesinde API ve MVC projeleri aynı anda çalıştırılabilmektedir.
- **Giriş Bilgileri:** Sisteme yönetici girişi yapmak için veritabanı oluşturulurken otomatik eklenen `admin@adesea.com` adresi ve `Admin123!` şifresi kullanılabilmektedir.

5. GELİŞTİRME SÜRECİ VE EKİP ÇALIŞMASI

AdeSea projesi, geliştirme sürecinde işbölümü yapılarak ortaklaşa hazırlanmıştır. Kod çakışmalarını önlemek amacıyla proje baştan API ve MVC olmak üzere iki ana bileşene ayrılmıştır.

API (Backend) Süreçleri: Veritabanı tasarımı, iş kurallarının yazılması ve servislerin oluşturulması süreçleri yürütülmüştür.

MVC (Frontend) Süreçleri: Kullanıcı arayüzü tasarımlarının kodlanması ve arayüzün API ile haberleştirilmesi işlemleri tamamlanmıştır.

Görev dağılımı bağımsız bir yapıda görünse de, ortak geliştirme noktalarında işbirliği yapılmıştır. Raporlama için gereken bazı sorguların hazırlanması ve SignalR altyapısının frontend entegrasyonu sürecinde çapraz çalışmalar gerçekleştirilmiştir. Geliştirilen API uç noktaları Swagger dokümantasyonu üzerinden izlenmiştir. Bu planlama sayesinde geliştirme süreci çakışma yaşanmadan tamamlanmıştır.

6. BENİM KATKILARIM (Adem Eraslan)

Projede MVC (Frontend) ağırlıklı çalışılmasının yanı sıra API (Backend) katmanında da önemli görevler üstlenilmiştir. Bu kapsamda yapılan başlıca çalışmalar şunlardır:

API ve Asenkron Bileşen Destekleri: Sadece ön yüz tasarımıyla kalınmayıp, admin panelindeki dinamik içerik yönetimi (CMS) ve destek talepleri (Ticket) modüllerinin API uç noktaları tarafımda kodlanmıştır. Grafikler ve yat kataloğu için gereken karmaşık okuma sorguları Dapper kullanılarak bizzat backend tarafında yazılmıştır.

Yapay Zeka Entegrasyonu: Sistemin yenilikçi özelliklerinden olan ve Google Gemini API üzerinden beslenen yapay zeka satış asistanının (Chatbot) backend servis entegrasyonu ve MVC arayüz haberleşmesi uçtan uca kurgulanmıştır.

Arayüz Tasarımı ve SignalR İstemcisi: HTML, CSS ve Bootstrap 5 kullanılarak modern sayfa yapıları oluşturulmuştur. SignalR altyapısı frontend tarafında dinlenerek üretim sürecinin anlık izlendiği görsel bileşen (Timeline) tasarlanmıştır. Ayrıca sistem loglarını izleme ekranı arayüze entegre edilmiştir.

Yat Tasarım Sihirbazı ve Kütüphaneler: Kullanıcının sayfa yenilenmeden yat konfigürasyonunu tamamlaması ve fiyatın güncellenmesi JavaScript ile sağlanmıştır. Dış servislere bağlı kalmadan doğrudan QR kod üreten QRCoder kütüphanesi ve yat satışlarını görselleştiren Chart.js kütüphanesi sisteme başarıyla entegre edilmiştir.

7. SONUÇ

AdeSea projesi ile lüks yat üretim ve sipariş süreçleri dijital bir platforma başarılı bir şekilde taşınmıştır. Projenin API ve MVC olarak ayrıştırılması, geliştirme sürecinin kesintisiz yürütülmesine olanak sağlamıştır.

- Proje sonucunda elde edilen kazanımlar şunlardır:
- Entity Framework ve Dapper kullanılarak veri güvenliği ve performans dengesi sağlanmıştır.
- Arka plan servisleri ve SignalR kullanımı ile asenkron işlemler pratik olarak uygulanmıştır.
- Yapay zeka entegrasyonu eklenerek modern yazılım gereksinimleri projeye dahil edilmiştir.
- Takım çalışması, kod düzeni ve doğru mimari planlamanın yazılım projelerindeki önemi net bir şekilde deneyimlenmiştir.

8. KAYNAKÇA

- Microsoft Learn - ASP.NET Core Web API ve MVC Belgeleri
- Microsoft Learn - Entity Framework Core Belgeleri
- Microsoft Learn - ASP.NET Core SignalR Belgeleri
- GitHub - Dapper Resmi Dokümantasyonu
- Serilog Dokümantasyonu
- QuestPDF Resmi Sayfası