

**softITo 4. DÖNEM 1. GRUP
BİLİŞİM YAZILIM AKADEMİSİ
BİTİRME PROJESİ RAPORU**



HERMES ERP VE PORTFÖY YÖNETİM SİSTEMİ

Rol Bazlı Erişim Kontrollü (RBAC) Kurumsal Kaynak Planlama Paneli

Beyza Mensur

İÇİNDEKİLER

1. GİRİŞ	3
1.1. Projenin Amacı	3
1.2. Proje Kısıtları	3
2. MATERYAL VE YÖNTEM	4
2.1. React.js ve Modern Bileşen Mimarisi	4
2.2. Redux Toolkit ve Durum Yönetimi	4
2.3. Tailwind CSS ve Arayüz Tasarımı	4
2.4. JSON Server ve RESTful API Simülasyonu	4
2.5. Recharts ile Veri Görselleştirme	4
3. KURULUM	5
3.1. Geliştirme Ortamı ve Kurulum Adımları	5
3.2. Uygulama Modülleri ve Ekran Tasarımları	5
3.2.1. Kullanıcı Giriş Modülü (Login)	5
3.2.2. Yönetici Paneli (Admin Dashboard)	6
3.2.3. Muhasebe ve Finans Modülü	7
3.2.4. Depo ve Stok Yönetimi Modülü	8
3.2.5. İnsan Kaynakları Modülü	9
3.2.6. Raporlama Paneli ve Grafikler	10
3.2.7. Sistem Günlükleri (Logs)	10
3.2.8. Rol Bazlı Yetkilendirme ve Korumalı Rotalar	11
3.2.9. Test ve Doğrulama Adımları	11
4. SONUÇ	12
5. KAYNAKÇA	12

1. GİRİŞ

Küreselleşen pazar koşullarında ve hızlı dijital dönüşüm süreçlerinde, ticari işletmelerin operasyonel verimliliklerini artırmaları ve kaynaklarını optimize etmeleri kritik bir öneme sahiptir. Özellikle teknoloji bileşenleri dağıtımı yapan, yüksek hacimli envanter ve değişken maliyet yapılarına sahip şirketlerde; ürün stoklarının, finansal işlemlerin, çalışan verilerinin ve lojistik hareketlerin senkronize bir şekilde yönetilmesi gerekmektedir. Kurumsal Kaynak Planlama (ERP) sistemleri, bu entegrasyonu sağlayan temel yazılım altyapılarıdır.

Bu proje kapsamında geliştirilen **Hermes ERP ve Portföy Yönetim Sistemi**, teknoloji bileşenleri dağıtımı alanında faaliyet gösteren orta ve büyük ölçekli bir firmanın tüm departman içi ve departmanlar arası operasyonlarını tek bir çatı altında toplamaktadır. Sistem, modern web teknolojileri ve rol bazlı erişim denetimi (RBAC) ilkeleri çerçevesinde tasarlanmıştır.

1.1. Projenin Amacı

Bu projenin temel amacı, yetkilendirilmiş personellerin kendi sorumluluk alanlarına göre özelleştirilmiş arayüzlere eriştiği, veri güvenliğinin en üst düzeyde korunduğu, gerçek zamanlı stok ve finansal maliyet hesaplamalarının yapıldığı modern ve kurumsal bir ERP yazılım mimarisi ortaya koymaktır.

Sistem; envantere eklenen yeni ürünlerin ağırlıklı ortalama maliyetlerini otomatik olarak hesaplamakta, departmanların harcamalarını ve nakit akış grafiklerini dinamik olarak güncellemekte, personellerin izin durumlarını ve organizasyon hiyerarşisini listelemekte ve sistemde gerçekleştirilen her kritik işlemi güvenlik amacıyla log altına almaktadır.

1.2. Proje Kısıtları

Proje geliştirme sürecinde ve mimari tasarım aşamasında aşağıdaki teknik ve operasyonel kısıtlar göz önünde bulundurulmuştur:

- Kısıt 1: İstemci Tabanlı Güvenlik Yetkilendirmesi (RBAC):** Uygulama, istemci tarafında (frontend) çalışacak bir tek sayfa uygulaması (SPA) olarak geliştirilmiştir. Yetkilendirme kontrolleri sunucu yerine yerel depolama (localStorage), Redux durumu ve React rotaları (Protected Routes) üzerinden yürütülmektedir.
- Kısıt 2: Yerel Veritabanı ve API Simülasyonu:** Gerçek zamanlı bir SQL/NoSQL veritabanı sunucusu yerine, hafif yapılı ve RESTful API simülasyonu sağlayan bir JSON Server altyapısı (db.json) kullanılmıştır.
- Kısıt 3: Ağ ve Dağıtım Sınırları:** Sistem lokal geliştirme ortamında çalışmak üzere yapılandırılmış olup, dağıtık veritabanı senkronizasyonu veya harici kimlik sağlayıcı (OAuth/Single Sign-On) entegrasyonları kapsam dışı bırakılmıştır.

2. MATERYAL VE YÖNTEM

Hermes ERP sisteminin mimarisi, ölçeklenebilir, modüler ve yüksek performanslı modern web teknolojileri üzerine inşa edilmiştir. Uygulama, frontend katmanı olarak React.js, global durum yönetimi olarak Redux Toolkit, stil katmanı olarak Tailwind CSS, veri saklama katmanı olarak JSON Server ve veri görselleştirme katmanı olarak Recharts kütüphanelerini kullanmaktadır.

2.1. React.js ve Modern Bileşen Mimarisi

Kullanıcı arayüzlerinin oluşturulmasında bildirimsel (declarative) ve bileşen tabanlı (component-based) yapısı nedeniyle React.js tercih edilmiştir. Modüller (Muhasebe, Depo, İK) bağımsız bileşenler halinde yazılmış olup, Sanal DOM (Virtual DOM) teknolojisi sayesinde veri güncellemelerinde sadece değişen arayüz elemanlarının yeniden çizilmesi (re-rendering) sağlanmış, bu da akıcı bir kullanıcı deneyimi sunmuştur.

2.2. Redux Toolkit ve Durum Yönetimi

Uygulama genelinde kullanıcı oturum bilgileri, tema tercihleri, envanter hareketleri, departman verileri ve finansal işlemler gibi birçok bileşenin ortak eriştiği veriler bulunmaktadır. Bu verilerin karmaşık bileşen ağaçlarında güvenli ve senkronize bir şekilde taşınabilmesi amacıyla Redux Toolkit (RTK) kullanılmıştır. RTK'in `createSlice` ve `createAsyncThunk` yapıları ile asenkron API istekleri yönetilmiştir.

2.3. Tailwind CSS ve Arayüz Tasarımı

Tasarımın kurumsal, modern ve "Enterprise" düzeyde olabilmesi için Tailwind CSS kullanılmıştır. Utility-first (hizmet odaklı) CSS yaklaşımı sayesinde hızlı prototipleme yapılmış, saydam cam efekti (glassmorphism), yumuşak border-radius değerleri, renk kodlu rozetler (badges) ve geçiş efektleri gibi üst düzey görsel estetikler CSS karmaşasına girilmeden uygulanmıştır. Ayrıca sistemde karanlık (dark) ve aydınlık (light) mod desteği yer almaktadır.

2.4. JSON Server ve RESTful API Simülasyonu

Uygulamanın veri saklama, okuma ve güncelleme ihtiyaçları için JSON Server kullanılmıştır. Proje kök dizininde yer alan `db.json` dosyası, standart bir REST API gibi çalışarak `/kullanicilar`, `/urunler`, `/personeller`, `/giderler` ve `/loglar` uç noktaları (endpoints) üzerinden CRUD operasyonlarına tam destek sunmaktadır.

2.5. Recharts ile Veri Görselleştirme

Finansal ve operasyonel metriklerin hızlıca analiz edilebilmesi amacıyla, React için optimize edilmiş SVG tabanlı Recharts grafik kütüphanesi kullanılmıştır. Yönetici panelinde gelir-gider trendleri çizgi grafiğiyle (Line Chart), raporlar panelinde ise stok değer dağılımları dilim grafiğiyle (Pie Chart)

3. KURULUM

3.1. Geliştirme Ortamı ve Kurulum Adımları

Projenin yerel geliştirme ortamında ayağa kaldırılması ve çalıştırılması için aşağıdaki kurulum adımları izlenmiştir:

- Node.js Yüklenmesi:** Uygulama JavaScript çalışma ortamı olarak Node.js (v18+) gerektirmektedir.
- Bağımlılıkların Kurulumu:** Proje kök dizininde terminalden `npm install` komutu verilerek `package.json` dosyasındaki tüm paketler kurulmuştur.
- Veri Sunucusunun Başlatılması:** JSON Server'ı çalıştırmak için aşağıdaki komut koşturulur:

```
npm run server
```

Bu komut veritabanını 3001 portunda aktif hale getirir.

- Frontend Sunucusunun Başlatılması:** Vite geliştirme sunucusunu başlatmak için aşağıdaki komut koşturulur:

```
npm run dev
```

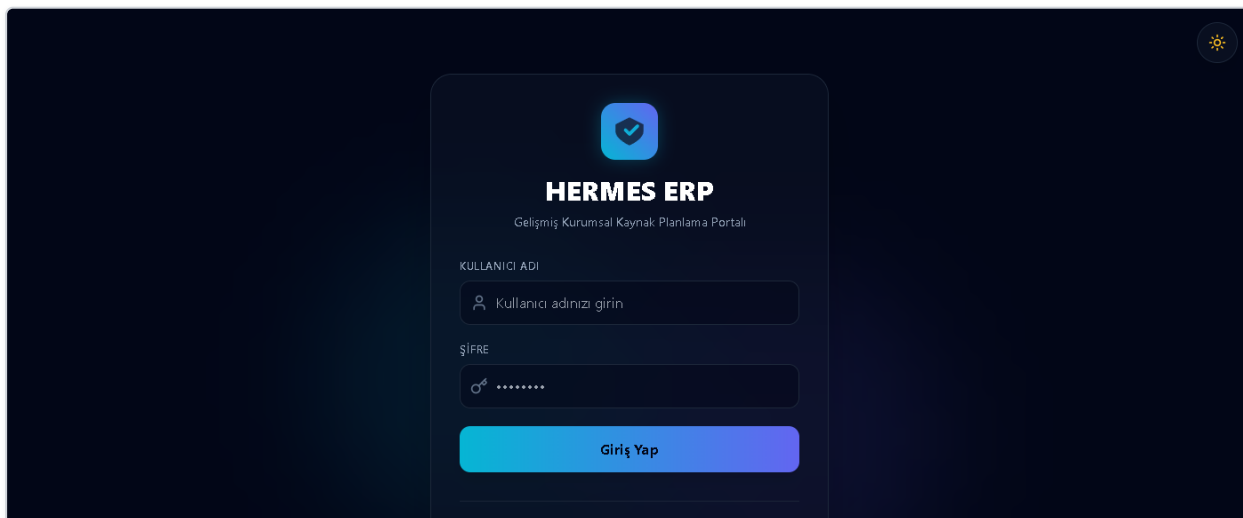
Bu komut ile istemci arayüzü `http://localhost:3000` adresinde yayına başlar.

3.2. Uygulama Modülleri ve Ekran Tasarımları

Bu bölümde, uygulamanın ana modülleri, işlevleri ve sisteme ait ekran görüntüleri detaylandırılmıştır.

3.2.1. Kullanıcı Giriş Modülü (Login)

Sisteme yetkisiz erişimi engellemek amacıyla tasarlanan giriş ekranı, modern cam tasarımı ve arka plan ışıklandırmaları içermektedir. Kullanıcılar kullanıcı adı ve şifreleri ile giriş yapabilmektedir. Ayrıca test süreçlerini kolaylaştırmak amacıyla sayfanın altında "Hızlı Giriş" butonları yer almaktadır.

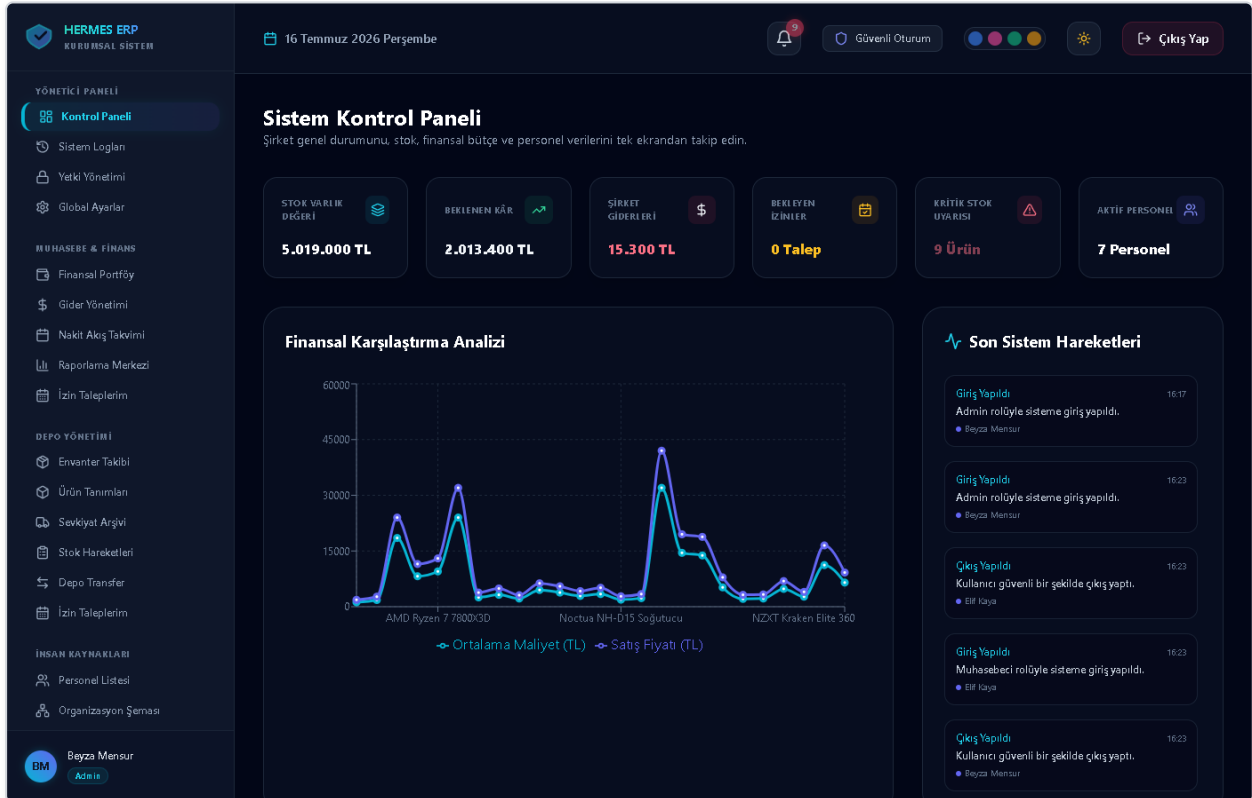


3.2.2. Yönetici Paneli (Admin Dashboard)

Sisteme "Admin" rolü ile giriş yapıldığında karşılaşılan bu panel, tüm organizasyonun genel durumunu tek bir ekranda özetlemektedir. Panelde üç adet yüksek görünürlüklü KPI (Temel Performans Göstergesi) kartı ve bir adet etkileşimli finansal analiz grafiği yer almaktadır:

- **Toplam Stok Değeri:** Depoda bulunan tüm teknolojik bileşenlerin miktarları ve ağırlıklı ortalama maliyetleri çarpılarak hesaplanan toplam finansal envanter değerini gösterir.
- **Güncel Kar/Zarar:** Satış fiyatları ile maliyet fiyatları arasındaki farklardan oluşan toplam brüt kâr değerini temsil eder.
- **Aktif Personel Sayısı:** Şirket bünyesinde çalışan güncel personel sayısını listeler.
- **Finansal Trend Çizgisi:** Recharts kullanılarak çizilen bu grafik, aylık bazda gelir, gider ve net kâr dağılımını görselleştirir.

Giriş yapan kullanıcının "Admin" rolü, sol menüdeki tüm departman sekmelerinin (Muhasebe, Depo, İK, Loglar, Ayarlar) aktif olmasını sağlar.



Şekil 3.2.2.1. Yönetici Paneli Arayüzü (Dashboard.jsx)

3.2.3. Muhasebe ve Finans Modülü

Muhasebe modülü, finans sorumlularının envanter maliyetlerini, faturaları, şirket giderlerini ve nakit akışını izlediği gelişmiş bir alandır. Modülün en kritik işlevi, ürün bazlı **Ağırlıklı Ortalama Maliyet** hesaplamasıdır.

Sisteme yeni bir ürün alımı girildiğinde veya mevcut bir ürüne ekleme yapıldığında, muhasebe modülü otomatik olarak şu formülü çalıştırır:

$$\text{Ağırlıklı Ortalama Maliyet} = \text{Toplam Satın Alma Maliyeti} / \text{Toplam Ürün Miktarı}$$

Hesaplanan bu maliyet değeri, güncel satış fiyatı ile karşılaştırılarak her bir ürün kalemi için kâr/zarar yüzdesi ve tutarı dinamik olarak tabloda listelenir. Tablodaki veriler renk kodlu badge'ler ile (kâr durumunda yeşil, zarar durumunda kırmızı) gösterilir. Muhasebe departmanı ayrıca "Giderler" ve "Nakit Akış" sayfalarından ek veri girişleri yapabilir.

Muhasebe ve Finans Paneli

Ağırlıklı ortalama maliyet hesaplamalarını, kârlılık durumunu ve tedarikçileri yönetin.

Finansal Ürün Portföyü

ÜRÜN ADI	TEDARİKÇİ	STOK	AĞ. ORT. MALİYET	SATIŞ FİYATI	KÂR ORANI
DDR5 RAM 16GB Bar kod: 8680001110015	Arena Bilgisayar	110 Adet	1.200 TL	1.850 TL	%54
1TB NVMe M.2 SSD Bar kod: 8680001110022	İndeks Bilgisayar	2 Adet	1.800 TL	2.700 TL	%50
RTX 4070 Ekran Kartı Bar kod: 8680001110039	Penta Teknoloji	10 Adet	18.500 TL	24.000 TL	%30
Intel Core i7 İşlemci Bar kod: 8680001110046	Arena Bilgisayar	40 Adet	8.200 TL	11.500 TL	%40
AMD Ryzen 7 7800X3D Bar kod: 8680001110050	İndeks Bilgisayar	200 Adet	9.500 TL	13.000 TL	%37
ASUS ROG Swift OLED Monitör Bar kod: 8680001110067	Penta Teknoloji	12 Adet	24.000 TL	32.000 TL	%33
Logitech G Pro X Superlight Mouse Bar kod: 8680001110074	Arena Bilgisayar	65 Adet	2.500 TL	3.800 TL	%52

Yeni Alım / Fiyat Girişi

HIZLI KAYIT (MEVCUT ÜRÜN SEÇ)

- DDR5 RAM 16GB
- 1TB NVMe M.2 SSD
- RTX 4070 Ekran Kartı
- Intel Core i7 İşlemci
- AMD Ryzen 7 7800X3D
- ASUS ROG Swift OLED Monitör
- Logitech G Pro X Superlight Mouse
- Corsair K70 RGB Klavye
- Seagate IronWolf 4TB HDD
- MSI B650 Tomahawk Anakart
- Samsung 990 Pro 2TB SSD
- EVGA 850W Power Supply
- Lian Li O11 Dynamic Kasa
- Noctua NH-D15 Soğutucu
- Kingston Fury Beast 32GB DDR5
- Gigabyte RTX 4080 Super
- Intel Core i9 14900K
- AMD Ryzen 9 7950X
- SteelSeries Arctis Nova Pro
- WD Black SN850X 1TB
- ASUS TUF Gaming 750W PSU

Şekil 3.2.3.1. Muhasebe Maliyet Analiz Ekranı (Accounting.jsx)

3.2.4. Depo ve Stok Yönetimi Modülü

Depo sorumluları (Depocu) tarafından kullanılan bu modül, fiziksel ürün envanterinin takibi, yeni ürün girişleri, barkod basımı ve depolar arası ürün transfer süreçlerini yönetmektedir.

Modülün temel özellikleri şunlardır:

- **Envanter Takibi:** Ürün adı, barkod numarası, stok miktarı, kategori ve raf konumu bilgileri listelenir.
- **Stok Giriş/Çıkış Operasyonları:** Depoya yeni bir teknolojik ürün geldiğinde adet ve alış maliyetiyle sisteme eklenir. Bu ekleme işlemi anında Muhasebe tablosundaki ağırlıklı ortalama maliyeti tetikler.
- **Depolar Arası Transfer:** Farklı lokasyonlardaki depolar arasında stok kaydırmaları yapılmasını sağlar ve bu işlemler /transferler API endpoint'ine kaydedilir.

Depocu rolüyle giriş yapan bir kullanıcı, veri güvenliği kısıtları nedeniyle finansal kâr/zarar oranlarını ve personel maaşlarını göremez, sol menüde Muhasebe ve İK sekmeleri filtrelenerek gizlenir.

HERMES ERP
KURUMSAL SİSTEM

16 Temmuz 2026 Perşembe

Güvenli Oturum

Çıkış Yap

Depo Envanter Takip Merkezi

Barkod yönetimi, stok durumları ve envanter sevk işlemlerini gerçekleştirin.

Mevcut Stok Durumu

Tümü Kritik (5) Güvenli

Ürün adı veya barkod ile arayın...

ÜRÜN ADI	BARKOD	DEPODAKİ MİKTAR	DURUM
DDR5 RAM 16GB	8680001110015	110 Adet	Güvenli
1TB NVMe M.2 SSD	8680001110022	2 Adet	Kritik
RTX 4070 Ekran Kartı	8680001110039	10 Adet	Kritik
Intel Core i7 İşlemci	8680001110046	40 Adet	Güvenli
AMD Ryzen 7 7800X3D	8680001110050	200 Adet	Güvenli
ASUS ROG Swift OLED Monitör	8680001110067	12 Adet	Kritik
Logitech G Pro X Superlight Mouse	8680001110074	65 Adet	Güvenli

Toplam 25 üründen 1 - 7 arası gösteriliyor

Önceki 1 2 3 4 Sonraki

Stok Hareketi Girişi

Stok hareketi gerçekleştirmek için tablodan bir ürün seçin.

Şekil 3.2.4.1. Depo Envanter ve Stok Arayüzü (Inventory.jsx)

3.2.5. İnsan Kaynakları Modülü

İnsan Kaynakları (İK) sorumlularının çalışanları, organizasyon şemalarını ve izin süreçlerini yönettiği bölümdür. Modül, işletme organizasyon yapısını hiyerarşik ve şematik olarak sunar:

- **Personel Listesi:** Çalışanların adı, soyadı, departmanı, unvanı, işe giriş tarihi, e-posta adresi ve durum bilgileri yer alır.
- **Organizasyon Şeması (Org Chart):** Departman yöneticileri ve onlara bağlı personeller görsel bir ağaç yapısında sunulur kurumsal hiyerarşi netleştirilir.
- **İzin Yönetimi:** Personellerin yıllık izin, hastalık izni gibi talepleri İK paneline düşer. İK yöneticisi bu talepleri onaylama veya reddetme yetkisine sahiptir.

Bu panel sayesinde İK personeli, şirketin departman bazlı çalışan dağılımını organize ederken depodaki ürün miktarları gibi ilgisiz alanlara müdahale edemez.

PERSONEL	DEPARTMAN	ROL / YETKİ	İLETİŞİM	İŞLEM
Beyza Mensur	Yönetim	Admin	beyzamsur008@gmail.com 0555 111 2233	
Elif Kaya	Muhasebe	Muhasebeci	elif@sirket.com 0555 222 3344	
Fatma Şen	Muhasebe	Muhasebeci	fatma@sirket.com 0555 222 3355	
Can Demir	Lojistik	Depocu	can@sirket.com 0555 333 4455	
Murat Koç	Lojistik	Depocu	murat@sirket.com 0555 333 4466	
Ahmet Yılmaz	İnsan Kaynakları	İK	ahmet@sirket.com 0555 444 5566	
Ayşe Çelik	İnsan Kaynakları	İK	ayse@sirket.com 0555 444 5577	

Şekil 3.2.5.1. İnsan Kaynakları Personel Yönetim Arayüzü (HR.jsx)

3.2.8. Rol Bazlı Yetkilendirme ve Korunmalı Rotalar

Uygulamada güvenliği sağlamak için `react-router-dom` ile entegre çalışan bir `ProtectedRoute` bileşeni geliştirilmiştir. Bu bileşen, kullanıcının oturum açıp açmadığını ve rolünün ilgili sayfaya erişmeye yetip yetmediğini kontrol eder:

```
import { Navigate } from "react-router-dom"
import { useSelector } from "react-redux"

export default function ProtectedRoute({ children, allowedRoles }) {
  const { user, isAuthenticated } = useSelector((state) => state.auth)

  if (!isAuthenticated) {
    return <Navigate to="/login" replace />
  }

  if (allowedRoles && !allowedRoles.includes(user.rol)) {
    return <Navigate to="/" replace />
  }

  return children
}
```

Bu yapı sayesinde, örneğin bir Depocu doğrudan URL satırından `/muhassebe` adresine gitmeye çalışırsa, sistem yetkisiz erişimi algılayarak kullanıcıyı ana sayfaya yönlendirir.

3.2.9. Test ve Doğrulama Adımları

Sistemin kararlılığı ve doğruluğu aşağıdaki test senaryoları ile doğrulanmıştır:

- **Yetkisiz Erişim Testi:** Giriş yapmamış bir kullanıcının doğrudan tarayıcı çubuğundan rotalara erişimi engellenmiş ve `/login` sayfasına yönlendirilmiştir.
- **Rol Filtreleme Testi:** Depocu hesabıyla giriş yapıldığında sol menüde Muhasebe ve İK sekmelerinin render edilmediği doğrulanmıştır.
- **Maliyet Algoritması Testi:** Adedi 10, birim fiyatı 100 TL olan bir ürüne, depocu tarafından adedi 5, birim fiyatı 130 TL olan yeni stok girişi yapılmış; muhassebe tablosunda ağırlıklı ortalama maliyetin $(1000 + 650) / 15 = 110$ TL olarak tam doğrulukla güncellendiği teyit edilmiştir.

4. SONUÇ

Bu projede, teknoloji bileşenleri dağıtımı yapan bir işletmenin operasyonel süreçlerini tek bir merkezden güvenli bir şekilde yönetmesini sağlayan **Hermes ERP ve Portföy Yönetim Sistemi** başarıyla geliştirilmiştir.

React.js bileşen mimarisi ve Tailwind CSS'in esnekliği birleşerek kullanıcılara hızlı, akıcı ve görsel açıdan zengin bir arayüz sunulmuştur. Redux Toolkit entegrasyonu sayesinde departmanlar arası veri senkronizasyonu kararlı bir şekilde yürütülmüştür. Geliştirilen ağırlıklı ortalama maliyet hesaplama algoritması ile envanter alımlarının kârlılık üzerindeki etkisi anlık olarak raporlanabilmektedir.

Sonuç olarak, rol bazlı erişim denetimi (RBAC) ile donatılmış, veri güvenliğini ön planda tutan, modern standartlarda ve genişletilebilir bir ERP Dashboard prototipi tamamlanarak başarıyla test edilmiştir.

5. KAYNAKÇA

- [1] React.js Resmi Dokümantasyonu: <https://react.dev>
- [2] Redux Toolkit Resmi Kılavuzu: <https://redux-toolkit.js.org>
- [3] Tailwind CSS Dokümantasyonu: <https://tailwindcss.com>
- [4] Recharts Grafik Kütüphanesi Kılavuzu: <https://recharts.org>
- [5] JSON Server Resmi GitHub Deposu: <https://github.com/typicode/json-server>

