



**KOCAELİ ÜNİVERSİTESİ
KOCAELİ MESLEK YÜKSEKOKULU**

BİLGİSAYAR PROGRAMCILIĞI PROGRAMI

PROJE

PAKET SERVİS ARAÇ TAKİP

YASİN SUBAŞI

MAYIS 2021

ÖNSÖZ VE TEŞEKKÜR

Bitirme projesi kapsamında fikir ve görüşleri ile yol gösteren ve yardımlarını esirgemeyen saygıdeğer hocam Sayın Mustafa OF’a teşekkürü bir borç bilirim.

Mayıs – 2021

Yasin SUBAŞI

İÇİNDEKİLER

İÇİNDEKİLER.....	iii
ŞEKİLLER DİZİNİ.....	v
SİMGELER VE KISALTMALAR DİZİNİ.....	vi
Kısaltmalar	vi
PAKET SERVİS ARAÇ TAKİP	vii
ÖZET.....	vii
PACKAGE SERVICE VEHICLE TRACKING.....	viii
ABSTRACT.....	viii
GİRİŞ	1
2. MATERYALLER VE YÖNTEMLER	2
2.1. Veri Tabanı Sistemleri ve MS SQL.....	2
2.1.1. Veri tabanı tablo ilişkileri diyagramı	2
2.2. Microsoft Visual Studio	3
2.2.1. Microsoft Visual Studio geliştirme ortamında C#	3
2.3. C#.....	4
2.4. ASP.NET MVC.....	4
2.4.1. Entity Framework Core	4
2.4.2. Razor	5
2.5. Web Arayüz Tasarımı	5
2.5.1. JavaScript ve jQuery	5
2.5.2. Bootstrap.....	6
3. PAKET SERVİS ARAÇ TAKİP UYGULAMASI İÇERİK VE KULLANIMI	7
3.1. Giriş Sayfası.....	7
3.2. Kayıtlı Araçları Yönetme ve Görevlendirme Alanları	7
3.2.1. Yeni araç ekleme	8
3.2.2. Var olan aracı düzenleme.....	8
3.2.3. Araç silme	9
3.2.4. Aracı görevlendirme	10
3.2.4.1. Önceden yüklenmiş ürünleri işleme ve gösterme alt yapısı	10
3.2.4.2. Yükleme işlemi sırasında kullanılan değer hesaplama sistemi.....	11
3.2.4.3. Yükleme işlemi sonrasında hızlı rapor	14
3.2.4.3.1. Hızlı rapor için fiyat hesaplanması	15
3.2.4.4. Yüklenmiş aracın görevini sonlandırma.....	15

3.2.5. Kayıtlı ürünleri yönetme.....	18
3.2.5.1. Ürün ekleme.....	18
3.2.5.2. Ürün düzenleme	19
3.2.5.3. Ürün silme	20
3.2.6. Rapor sistemi ve z raporu.....	21
3.2.6.1. Seçme işlemlerinin alt yapısı ve veri gösterimi	25
3.2.6.2. Z raporu	27
3.2.6.2.1. Z raporunu popup olarak açma ve yazdırma sistemleri	27
3.2.7. Kullanıcı paneli ve profil yönetimi	28
3.2.7.1. Email yönetimi.....	28
3.2.7.2. Şifre yönetimi.....	29
3.2.7.3. Yeni kullanıcı oluşturma	30
3.2.7.4. Kişisel veri yönetimi	30
4. SONUÇLAR VE ÖNERİLER	32
KAYNAKLAR.....	33
EKLER.....	34
ÖZGEÇMİŞ	48

ŞEKİLLER DİZİNİ

Şekil 1 Veri Tabanına Ait İlişki Diyagramı	3
Şekil 2 Giriş Bölgesi	7
Şekil 3 Araçlar Sayfası	8
Şekil 4 Araç Düzenleme	9
Şekil 5 Araç Silme	9
Şekil 6 Araç Görevlendirme	10
Şekil 7 Önceden Yüklenmiş Ürünleri İşleyip Gösterime Sunmak İçin Kullanılan JavaScript Kodu	11
Şekil 8 Yüklenen Ürünlerin Toplam Değerini Hesaplayan JavaScript Kodu.....	12
Şekil 9 Seçilen Ürünleri Araca Yükleme İçin Kullanılan C# Kod Bloğu.....	14
Şekil 10 Yükleme İşlemi Sonrasında Hızlı Rapor	14
Şekil 11 Yükleme İşlemi Sonrasında Hızlı Rapor Fiyat Toplamı İçin JavaScript Kodu.....	15
Şekil 12 Görevi Sonlandırma Modali	16
Şekil 13 Aracın Görevini Sonlandırıp Rapora Yazma İşlemi İçin Kullanılan C# Kod Bloğu	17
Şekil 14 Ürünler Sayfası	18
Şekil 15 Ürün Oluşturma	19
Şekil 16 Ürün Düzenleme Sayfası	20
Şekil 17 Ürün Silme Sayfası	21
Şekil 18 Raporlar Sayfası	22
Şekil 19 Filtreleme, Sıralama İşlemini Yapan C# Kod Bloğu	24
Şekil 20 Rapor Kutucuğu Seçildiğinde Çalışan JavaScript Kodu.....	25
Şekil 21 Tümünü İşaretle Kutucuğu Seçildiğinde Çalışan JavaScript Kodu.....	26
Şekil 22 Seçilen Raporların Toplam Fiyatını Gösteren JavaScript Kodu.....	26
Şekil 23 Z Raporu	27
Şekil 24 Z Raporunu Popup Olarak Açmak İçin Kullanılan JavaScript Kodu.....	28
Şekil 25 Yönetici Paneli	28
Şekil 26 Email Yönetim Sayfası	29
Şekil 27 Şifre Yönetimi.....	29
Şekil 28 Yeni Kullanıcı Oluşturma	30
Şekil 29 Kişisel Veri Sayfası.....	31

SİMGELER VE KISALTMALAR DİZİNİ

@	: Razor Sözdizimi
{ }	: C# Kod Kapsamı
\$: JQuery Sözdizimi

Kısaltmalar

CSS	: Cascading Style Sheets (Basamaklı Stil Şablonları)
Db	: Database (Veri Tabanı)
EF	: Entity Framework (Varlık Çerçevesi)
GC	: Garbage Collector (Çöp Toplayıcı)
HTML	: Hypertext Markup Language (Hiper Metin İşaretleme Dili)
IDE	: Integrated Development Environment (Tümleşik Geliştirme Ortamı)
JS	: JavaScript
LING	: Language-Integrated Query (Dile Entegre Edilmiş Sorgu)
Ms	: Microsoft
MVC	: Model-View-Controller (Model-Görünüm-Denetleyici)
ORM	: Object Relational Mapping (Nesne İlişkisel Eşleme)
PSAÇ	: Paket Servis Araç Takip
SQL	: Structured Query Language (Yapılandırılmış Sorgu Dili)

PAKET SERVİS ARAÇ TAKİP

ÖZET

Günümüzde, çevrim içi alışverişin toplam ticaretten aldığı pay oldukça yüksek seviyelere ulaşmıştır. Bunun büyük bir bölümünü şehirler, ülkeler ve kıtalar arası büyük ölçekli transferler oluştursa da mikro ölçekteki bölgesel küçük işletmelerin bu ağlardan faydalanma imkânı yoktur. Bunun bir sonucu olarak bu grupta bulunan işletmeler kendi imkânlarıyla lojistik ağı oluşturmak durumunda kalırlar. Burada ortaya çıkan karmaşanın giderilmesi için bilişim teknolojilerini efektif olarak kullanılması bir zorunluluktur. Bu projede saha görevlileri için mobil ve ofis görevlileri için masaüstü erişimi destekleyen ASP.NET 3.1 MVC web teknolojisini kullanan, araca yükleme esnasından raporlamaya kadar tüm aşamalarda entegre çalışabilen bir sistem gerçekleştirilecektir.

Anahtar Kelimeler: ASP.NET, Lojistik, Paket Servis.

PACKAGE SERVICE VEHICLE TRACKING

ABSTRACT

Nowadays, the share of online shopping from total trade has reached quite high levels. Although large-scale transfers between cities, countries and continents make up a large part of this, micro-scale regional small businesses do not have the opportunity to benefit from these networks. As a result, businesses in this group have to create a logistics network with their own means. It is a necessity to use information technologies effectively in order to eliminate the confusion that arises here. In this project, a system that uses ASP.NET 3.1 MVC web technology, which supports mobile for field officers and desktop access for office staff, that can work in integration at all stages from loading to the vehicle, will be realized.

Keywords: ASP.NET, Logistics, Package Service.

GİRİŞ

Küçük ölçekli yerel hizmet sağlayıcıları lojistik operasyonlarını yönetme noktasında geçmişten günümüze birçok yola başvurmuşlardır. Değişen ve gelişen teknoloji günümüzde bu ihtiyaçların cross-platform olarak adlandırılan aynı türde olmayan fakat aynı görevleri yürütebilen sistemlerle giderilmesini şart koşturmaktadır.

Geliştirdiğimiz PSAÇ uygulaması web üzerinde çalışarak “cross-platform” yaklaşımını sonuna kadar kullanmaktadır. Uygulamamız saha görevlileri için mobil, ofis görevlileri için masaüstü erişimi destekleyen responsive bir tasarıma sahiptir. Uygulamada tasarım konusunda Bootstrap 5 ve jQuery teknolojilerinden, sunucu tarafında ise ASP.NET Core MVC teknolojisinden faydalanılmıştır. Projeyi kullanan işletmelerin lojistik operasyonlarını yürütmesine ek olarak geriye dönük raporlama ve analiz yapmalarının olanak sağlamak ve sistemin çalışması esnasında üretilen verileri saklamak için veri tabanı yönetim sistemi olarak Microsoft tarafından geliştirilen MS SQL tercih edilmiştir.

Proje temelinde işletmede satışı olan ürünlerin ve işletmenin kullanımında olan araçların bir alıcıyla eşlenerek lojistik karmaşasını ortadan kaldırmayı ve erişilebilir kılmayı aynı zamanda bunlara paralel olarak işletmenin yürütmesi gereken kayıt sorumluluklarını, arka planda raporlar ve z raporu özellikleriyle karşılayarak işletmeye direkt ve dolaylı olarak maddi tasarruf sağlamayı hedeflemektedir.

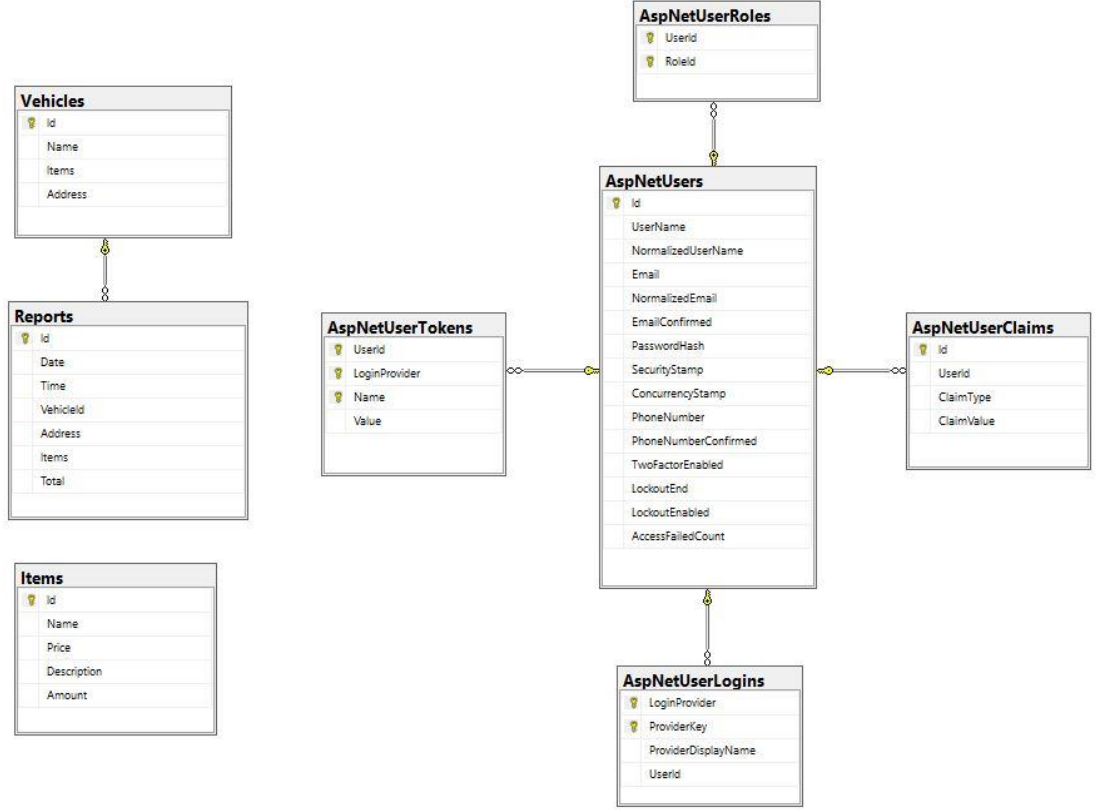
2. MATERYALLER VE YÖNTEMLER

2.1. Veri Tabanı Sistemleri ve MS SQL

Veri tabanı, sistemin devamlılığını sürdürebilmesi ve daha sonra gelebilecek potansiyel değişimlere ayak uydurabilmesi için ham verinin saklandığı yerdir. Veri tabanları kimi zaman işlevlerin temelindeki verileri barındırırken kimi zaman ise yalnızca arşiv niteliği taşıyan argümanları saklarlar, bazen ise kullanıcıya gösterilmek üzere bilgi verileri de tutabilirler. Veri tabanları ayrıca sistemlerin platformdan bağımsız olarak çalışmasını sağlarlar. MS SQL temelinde veri tabanları için bir standart hâline gelmiş SQL dilini kullanmaktadır. SQL dili veri tabanı üzerinde CRUD olarak anılan anlamlı veri oluşturma, oluşturulmuş verileri okuma, kayıtlı verileri güncelleme ve bu verileri silmeyi sağlar. Sayılan bu temel özelliklerin uygulanması noktasında verileri işleme maruz bırakabilecek özelliklere de sahiptir veri tabanı sistemlerinin temelinde SQL olsa da varyantlara ayrılmasının en temel gerekçelerinden birisi bu özelliklerin çeşitlendirilmesi ve verimli hâle getirilmesidir. Bu bağlamda Microsoft tarafından geliştirilen MS SQL teknolojisi, Microsoft'un tecrübelerinin bir yansıması olarak değerlendirilebilir. MS SQL; veri güvenliği, yedekleme, yönetimsel araç çeşitliliği, çoklu platform desteği özellikleriyle varyantlarından ayrılmaktadır.

2.1.1. Veri tabanı tablo ilişkileri diyagramı

Projede kullandığımız ilişkisel veri tabanımızı temsil eden diyagram şekil 1'de gösterilmiştir. Diyagram incelendiğinde fark edilebileceği üzere tablolar bire-bir ve bire-çok olarak ilişkilendirilmişlerdir.



Şekil 1 Veri tabanına ait ilişki diyagramı

2.2. Microsoft Visual Studio

Microsoft Visual Studio, Microsoft tarafından geliştirilen Microsoft'un kendi dilleri ve sistemlerini önceleyen çoklu platform desteğine sahip bir IDE'dir. Topluluk desteği ile kendi içerisinde eklentilerin paylaşılabildiği Marketplace sistemiyle birçok ek özellik sunmaktadır. Alternatif ortamlarda harici olarak sunulan IntelliSense (Akıllı Kod Tamamlama) özelliğini ise dahili olarak bulundurur. Ayrıca Visual Studio arka plan derleme özelliği ile de kod yazılırken arka planda derleme yaparak derleme hatalarını gösterir. Bununla birlikte projede kullandığımız MS SQL ile tamamen uyumludur ve kendi içinde proje ve veri tabanı bağlantısını uyumlu bir şekilde sağlar. [1]

2.2.1. Microsoft Visual Studio geliştirme ortamında C#

Nesneye yönelimli yazılım dillerinin başında gelen C#, Microsoft Visual Studio ile geliştirmesi ve hata ayıklaması oldukça kolay bir dil hâline gelmektedir. Visual Studio'nun gelişmiş hata ayıklama sistemi kodun çalışmasını adım adım gözden

geçirerek geliştiricinin belirlediği noktalarda sistemin tüm elemanları hakkında bilgi edinebilmeyi mümkün kılmaktadır. Tasarım yönüyle de alternatiflerinin arkasında kalmayan Microsoft Visual Studio C# için biçilmiş kaftandır. [2]

2.3. C#

.NET ekosistemi için özel olarak Microsoft tarafından geliştirilmiş, çoklu platform desteğine sahip, nesne yönelimli programlama dilidir. Otomatik GC özelliği ile bellek yönetimini kullanıcının ek bir işlem yapmasına gerek kalmadan kontrol eder. Genellikle liste değişken türlerinde kullanılan LINQ sorgularını içeren C# ile veride filtreleme, sıralama ve birleştirme gibi özelliklerden de faydalanılır. [3]

2.4. ASP.NET MVC

ASP.NET MVC, Microsoft tarafından ASP.NET üzerine inşa edilmiş Model-View-Controller desenini kullanan açık kaynaklı bir web sitesi geliştirme yapısıdır. Model, View ve Controller katmanlarından oluşan mimaride her katman kendisine adanmış spesifik bir işi yapmaya odaklıdır. Model katmanı veri tabanından gelen ham verinin formatlandığı katmandır. View katmanı kullanıcının gördüğü işlenmiş verilerin bir ürünü olan katmandır. Controller katmanı yalnızca formatlanmış verinin işlenerek View katmanına dönüştürüldüğü katmandır. Uygulamayı üç katmana bölerek karmaşıklığı ortadan kaldırmasının yanında HTML üzerinde tam kontrol sağlamasıyla da geliştiricilere kolaylık sağlar. [4]

2.4.1. Entity Framework Core

Entity Framework Core, geliştiricilere veri tabanındaki verileri depolamak ve bunlara erişmek için otomatikleştirilmiş mekanizma sunan açık kaynaklı ve platformlar arası bir ORM (Nesne İlişkisel Eşleyici) framework'üdür. EF Core, geliştiriciyi veri tabanı sorgularıyla zaman kaybettirmemek adına Model sınıflarını "DbContext" sınıfıyla eşleştirir. DbContext sınıfı ile Migration (Geçiş) oluşturulur. Migration, veri tabanını oluşturmaya ve Model değiştiğinde veri tabanını güncellemeye yarar. DbContext sınıfı sayesinde veri tabanından okuma ve yazma, veri tabanını sorgulama ve LINQ ile kullanarak filtreleme yapılabilir. [5]

2.4.2. Razor

Razor, sunucu tabanlı kodu C# ve VB.NET kullanarak web sayfalarına yerleştirmenize imkân tanıyan bir işaretleme sözdizimidir. Razor, bir programlama dilinden ziyade sunucu tarafı biçimlendirme dilidir. Razor'un ASP.NET MVC ile hiçbir bağı yoktur çünkü Razor genel amaçlı bir şablon oluşturma motorudur. Kendisine kaynak aldığı Razor formatında yazılmış HTML dosyalarını Controller tarafından kendisine getirilen verilere göre manipüle ederek kullanıcıya özel çıktı oluşturmak için çalışır. İşleri kolaylaştırmak adına ASP.NET, HTML üretmek için Razor kullanmamıza izin veren bir görünüm motorunu tercih etmiştir. Razor ile HTML sayfalarında C# kodu kullanabilir, yalnızca HTML dosyalarını manipüle etmenin yanında katmanlar arası veri taşıyabilir. Razorlar, ASP.NET MVC'nin .cshtml uzantılı View dosyalarında "@" etiketi ile başlayarak kullanılırlar. [6]

2.5. Web Arayüz Tasarımı

Web sitelerin arayüzleri için sektörün standartları hâline gelmiş HTML ve CSS dilleri kullanılır. HTML, bir dizi öğeden oluşur ve tarayıcıya içeriğin nasıl görüntülenmesi gerektiğini söyler. Bir HTML öğesi; başlangıç etiketi, içerik ve bitiş etiketiyle tanımlanır. HTML, normalde .html uzantılı dosyalarda yazılsa da ASP.NET'te Razor tarafından üretildiği için geliştirme sırasında .cshtml uzantılı olarak kullanılırlar. HTML bir arayüzün yapısını oluştururken, CSS bu HTML elementlerinin kullanıcıya nasıl gösterileceğini kontrol eder. CSS stil sayfaları .css uzantılı dosyalarda bulunur. CSS özelliklerinden daha çok Bootstrap'in manipülasyon gerektiren özelliklerini kullanılırken gerek duyuldu.

2.5.1. JavaScript ve jQuery

JavaScript, yorumlanmış bir programlama dilidir. HTML ile entegre olduğu için uygulaması çok kolaydır. JavaScript ile HTML sayfalarının statik yapısını hareketlendirmek ve sunucudan bağımsız yapılabilecek kullanıcı tarafı işlemlerin yürütülmesini sağlamak oldukça kolaydır. Bu işlemler projemizde kimi zaman fiyatın hesaplanıp statik HTML sayfasının kullanıcı tarafı olarak manipüle edilmesi için kimi zaman ise kullanıcı tercihleri doğrultusunda sitenin tasarımını yöneten CSS dosyalarının kullanıcı tarafındaki kopyalarını manipüle etmek için kullanılmıştır. JavaScript kodları .js uzantılı dosyalarda yazılıp HTML sayfasına "<script src="

“dosya yolu”/>” şeklinde eklenebilir ya da direkt olarak HTML içerisinde JavaScript kodu “<script>” ve “</script>” etiketlerinin arasında yazılabilir. Projede JavaScript çoğunlukla fiyat hesaplama işlemlerinin gösterilmesi için kullanılmıştır. jQuery, JavaScript kullanımını kolaylaştıran “daha az yaz, daha çok yap” prensipli bir JavaScript kütüphanesidir. JavaScript ile birçok satırda yapılabilecek işi tek bir kod satırıyla yapabilmemizi sağlar.

2.5.2. Bootstrap

İlk başlarda Twitter’daki geliştiricilerin kendi kişisel kullanımları için geliştirdikleri Bootstrap daha sonraları açık kaynak, esneklik ve kullanım kolaylığı ile geliştiriciler arasında popüler hâle gelmiştir. Bootstrap’in kendi içinde bulundurduğu CSS ve JavaScript dosyalarıyla HTML elementlerinin class niteliğine eklemeler yaparak görünüş elde edilebilir. Formlar, butonlar, tablolar, navigasyon ve modallar gibi birçok şablonu içerisinde bulundurur. Bootstrap ayrıca responsive yapısıyla telefon, tablet, masaüstü ve diğer modern tarayıcılarla uyumludur. Projede platformlar arası uyumluluk sağlamak ve modern tasarım elde etmek için Bootstrap kullanılmıştır. [7]

3. PAKET SERVİS ARAÇ TAKİP UYGULAMASI İÇERİK VE KULLANIMI

3.1. Giriş Sayfası

Herhangi bir oturum açmamış kullanıcı sadece ana sayfaya ve giriş yapma kısmına erişebilir. Sistem bir işletme sistemi olduğu için harici kayıt sayfası eklenmemiştir. Hesap işlemleri ve yönetimi Microsoft.AspNetCore.Identity.EntityFrameworkCore paketi ile yapılmıştır.



The image shows a login form with a large black circular icon containing a white person silhouette at the top. Below the icon are two input fields: the first is labeled 'E-Posta' and the second is labeled 'Şifre'. Below the password field is a checkbox labeled 'Beni hatırla'. At the bottom of the form is a blue button with the text 'Giriş Yap'.

Şekil 2 Giriş bölgesi

3.2. Kayıtlı Araçları Yönetme ve Görevlendirme Alanları

Sistemimizde araçlar sayfasında yeni araç ekleyebilir, var olan aracı düzenleyebilir, silebilir, araca yükleme yapılabilir veya varsa araçtaki yüklenmiş ürünler ve aracın ürünleri teslim edeceği adres görülebilir. Araca yükleme yapıldıktan sonra etkinleşen “Görevi Sonlandır” butonu ile de yüklemeler araçtan silinip raporlara yazılır.

Araçlar

Araç ekle

Araç No	İsim	Ürünler	Adres	Olaylar
6	Honda Cbf 150			Düzenle Sil Yükleme Yap Görevi Sonlandır
7	Haojue 135 Da			Düzenle Sil Yükleme Yap Görevi Sonlandır
8	Bajaj Pulsar 200 Ns			Düzenle Sil Yükleme Yap Görevi Sonlandır
9	Tvs Apache RTR 150			Düzenle Sil Yükleme Yap Görevi Sonlandır
10	Yamaha Ybr 125			Düzenle Sil Yükleme Yap Görevi Sonlandır

Şekil 3 Araçlar sayfası

3.2.1. Yeni araç ekleme

Araç oluşturma sayfasında araca spesifik bir ad tanımlayarak oluşturma işlemi hızla tamamlanabilir. Sistem tarafından geriye kalan bilgiler otomatik olarak tanımlanarak veri tabanına kayıt işlemleri gerçekleştirilecektir.

Oluştur

Araç

İsim

Oluştur

[Listeye Dön](#)

Şekil 4 Araç oluşturma

3.2.2. Var olan aracı düzenleme

Araç oluşturmada işlemindeki gibi kullanıcı tarafından yalnızca araca tanımlanmış spesifik isim düzenlenebilir. Aracın yüklendiği ürünler ve gönderildiği adres operasyon bazlı olduğu için araç düzenleme ekranında yer verilmez.

Düzenle

Araç

İsim

Honda Cbf 150

Kaydet

[Listeye Dön](#)

Şekil 5 Araç düzenleme

3.2.3. Araç silme

Araçların isimlendirilmesinde marka-model formatında adlandırma zorunluluğu bulunmasa da kullanıcılar tarafından adlandırma konusunda sık kullanılan bir format olduğu için araç silme ekranında kullanıcılar araçların adlarına ek olarak benzersiz kimlik numaralarını da görürler.

Silmek istediğinden emin misin?

Araç

Araç No:

6

İsim:

Honda Cbf 150

Sil

[Listeye Dön](#)

Şekil 6 Araç silme

3.2.4. Aracı görevlendirme

Araçların listelendiği sayfadan “Yükleme Yap” butonu kullanılarak erişilen bu sayfada sisteme kayıtlı tüm ürünler listelenir. Araç aktif olarak görevdeyken bu sayfa kullanılırsa yüklenecek ürünlerin veya ürünlerin gönderileceği adresin değiştirilmesini sağlar, görevde değilken ise aracı yeni bir göreve göndermek için ürünler seçilir ve adres tanımlanır. Ayrıca firmaların ürün çeşitliliğinin çok yüksek olması durumunda sayfa yapısının bozulmaması için ürünlerin listelendiği tablo “<div class=“overflow-auto table-height“ style=“height: 50vh;“>” ve “</div>” elementleri arasına alınarak Bootstrap ve CSS yardımıyla kaydırma özelliği eklenmiştir.

Honda Cbf 150 İsimli Araca Yükleme Yap

Yükleme yap

Adres:

Gölcük

Toplam ₺1950

Yüklensin mi?	Adet	İsim	Fiyat	Açıklama
<input checked="" type="checkbox"/>	<input type="text" value="100"/>	Ekmek	₺5	
<input checked="" type="checkbox"/>	<input type="text" value="100"/>	Ülker Çikolatalı Gofret	₺5	
<input checked="" type="checkbox"/>	<input type="text" value="100"/>	Dondurma	₺5,5	
<input checked="" type="checkbox"/>	<input type="text" value="100"/>	Çokomel	₺4	4'lü Paket
Yüklensin mi?	Adet	İsim	Fiyat	Açıklama

Listeye dön

Şekil 7 Araç görevlendirme

3.2.4.1. Önceden yüklenmiş ürünleri işleme ve gösterme alt yapısı

Görevlendirilmiş bir araç üzerindeki yükler düzenlenmek için yeniden düzenleme sekmesine alınırsa aracın üzerinde yüklü olan adres ve ürün bilgilerinin ön yüklü olarak gelmesini sağlayan kod parçasıdır.

```

function load(items) {
    let map = new Map();
    var split = items.split(" - ");
    for (var a = 0; a < split.length; a++) {
        var item = split[a];
        var itemSplit = item.split(" adet ");
        var amount = parseInt(itemSplit[0]);
        var itemName = itemSplit[1];
        map.set(itemName, amount);
    }
    var amounts = $("input[id$='__Amount']");
    for (var i = 0; i < amounts.length; i++) {
        var amount = amounts[i].value;
        if (!amount || parseInt(amount) < 1) {
            var name = document.getElementById("name " + i).innerText;
            amount = 1;
            if (map.has(name)) {
                amount = map.get(name);
                document.getElementById("checkbox " + i).checked = true
            }
            document.getElementById(i + "__Amount").value = amount;
        }
    }
}

```

Şekil 8 Önceden yüklenmiş ürünleri işleyip gösterime sunmak için kullanılan JavaScript kodu

3.2.4.2. Yükleme işlemi sırasında kullanılan değer hesaplama sistemi

Kullanıcının yükleme yapmak için seçtiği ürünlerin değerini hem kullanıcı için deneyimi aksatmayacak hem de sunucu için ek yük oluşturmayacak bir şekilde

hesaplamak için kullanıcı taraflı olarak çalışan JavaScript’i tercih ettim. Kullanıcının görüntülediği her veri satırını jQuery yardımıyla kontrol ederek eğer “Yüklensin mi?” alanı seçiliyse ürünün adet fiyatından kullanıcının girdiği adet kadar hesaplanan toplam fiyata eklenir. Ulaşılan son fiyat “document.getElementById(“text”)” metodu kullanılarak sayfada kendisine özel id ile tanımlanmış alana yazılır.

```
function calculate() {  
    var text = document.getElementById("text");  
    text.style.display = "block";  
    var amounts = $("input[id$='__Amount']");  
    var prices = $("td[id='price']");  
    var tot = 0.0;  
    for (var i = 0; i < amounts.length; i++) {  
        var amount = amounts[i].value;  
        if (!amount || parseInt(amount) < 1) {  
            amount = 1;  
            document.getElementById(i + "__Amount").value = amount;  
        }  
        var price = parseFloat(prices[i].innerHTML.replace('₺', ' ').replace(',', '.'));  
        if (document.getElementById("checkbox " + i).checked &&  
            parseInt(amount))  
            tot += parseInt(amount) * price;  
    }  
    text.innerHTML = "Toplam ₺" + tot;  
}
```

Şekil 9 Yüklenen ürünlerin toplam değerini hesaplayan JavaScript kodu

```

[HttpPost]

public ActionResult LoadItem(List<Item> model, int[] ids, String address)
{
    int vehicleId = (int)TempData["VehicleId"];
    Vehicle vehicle = dbContext.Vehicles.Single(x => x.Id == vehicleId);

    if (ids.Length < 1)
    {
        if(vehicle.Items == null)
        {
            ViewData["Error"] = "Hiç ürün seçmedin.";
            return View("Error");
        }
        else
        {
            vehicle.Items = null;
            vehicle.Address = null;

            dbContext.Entry(vehicle).State = EntityState.Modified;
            dbContext.SaveChanges();
            return View("Complete", null);
        }
    }
    else if(String.IsNullOrEmpty(address))
    {
        ViewData["Error"] = "Bir adres girmelisin.";
        return View("Error");
    }
    else
    {
        List<Item> selected = model.FindAll(item => ids.Contains(item.Id));
        double total = 0;

        foreach(Item item in selected)
        {
            if(item != null && item.Amount < 1)
            {

```

```

        ViewData["Error"] = "Adet 1'den küçük olamaz.";
        return View("Error");
    }

    total += item.Amount.Value * item.Price;
}

string items = Join(selected, " - ", " adet ");

vehicle.Items = items+"|"+total;
vehicle.Address = address;

dbContext.Entry(vehicle).State = EntityState.Modified;
dbContext.SaveChanges();

return View("Complete", selected);
}
}

```

Şekil 10 Seçilen ürünleri araca yüklemek için kullanılan C# kod bloğu

3.2.4.3. Yükleme işlemi sonrasında hızlı rapor

Bir araca yükleme yapıldıktan sonra işlem anında raporlara taşınmaz fakat personelin yaptığı işlemin sonucunu topluca görebilmesi için işlem sonunda hızlı rapor görüntülenir.

Belirtilen ürünler Honda Cbf 150 adlı araca yüklendi:

- 100 adet Ekmek = ₺500
- 100 adet Ülker Çikolatalı Gofret = ₺500
- 100 adet Dondurma = ₺550
- 100 adet Çokomel = ₺400

Toplam ₺1950

Yüklemeye dön

Listeye dön

Şekil 11 Yükleme işlemi sonrasında hızlı rapor

3.2.4.3.1. Hızlı rapor için fiyat hesaplanması

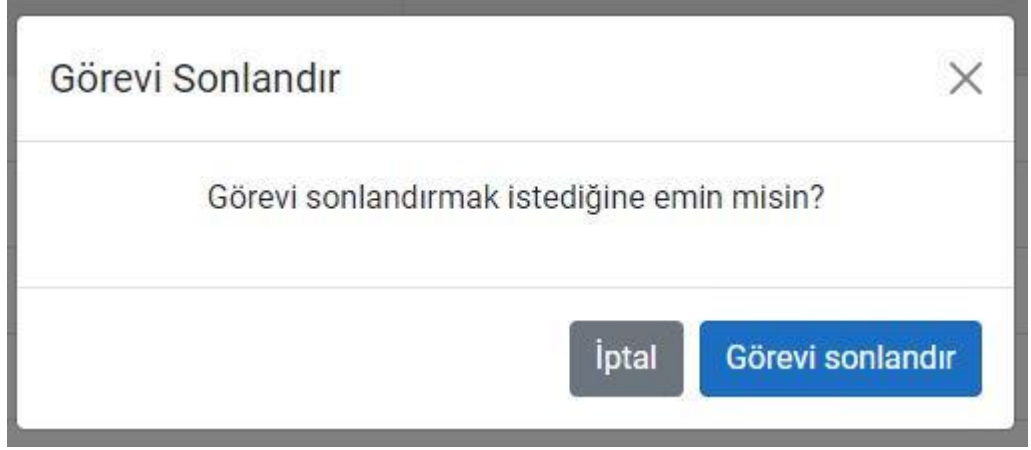
Bir araca ürün yüklendiğinde veya daha önce görevlendirilmiş aracın adres veya ürün bilgileri güncellendiğinde kullanıcıyı tüm bilgiler bir arada olarak bilgilendirmek için kullanılan hızlı rapor sayfasındaki ürünlerin toplam fiyatını hesaplayan kod parçası.

```
function calculateLoad() {  
    var elements = $("li[id^='entry']");  
    var total = 0.0;  
    for (var i = 0; i < elements.length; i++) {  
        var element = elements[i];  
        var price = parseFloat(element.getAttribute("price").replace(',', '.'));  
        var amount = parseInt(element.getAttribute("amount"));  
        var result = price * amount;  
        element.innerHTML += " = ₺" + result;  
        total += result;  
    }  
    document.getElementById("text").innerHTML = "Toplam ₺" + total;  
}
```

Şekil 12 Yükleme işlemi sonrasında hızlı rapor fiyat toplamı için JavaScript kodu

3.2.4.4. Yüklenmiş aracın görevini sonlandırma

Şekil 3.2’de yer alan soluk renkli “Görevi Sonlandır” butonu yüklenmiş araçlar için aktif diğer butonlar gibi gözükür. Kullanıcıların yanlışlıkla işlem yapmalarını engellemek için ek bir doğrulama sayfası görüntülenir. Bu işlem onaylandığında araca yüklenmiş ürünler ve aracın gönderildiği adres raporlar sayfasında görüntülenmeye başlar ve araç yeniden kullanım için hazır hâle getirilir.



Şekil 13 Görevi sonlandırma modalı


```

public IActionResult SaveItem(int id)
{
    Vehicle vehicle = dbContext.Vehicles.Single(x => x.Id == id);

    DateTime date = DateTime.Now;

    int hour = date.Minute;

    string hourAsString = hour.ToString().Length == 1 ? "0" + hour : hour.ToString();

    int minute = date.Minute;

    string minuteAsString = minute.ToString().Length == 1 ? "0" + minute : minute.ToString();

    string time = date.Hour + ":" + minuteAsString;

    string[] split = vehicle.Items.Split("|");

    double total = Double.Parse(split[1]);

    Report report = new Report()
    {
        Date = date,
        Time = time,
        VehicleId = vehicle.Id,
        Address = vehicle.Address,
        Items = split[0],
        Total = total
    };

    vehicle.Items = null;
    vehicle.Address = null;
    dbContext.Entry(vehicle).State = EntityState.Modified;
    dbContext.SaveChanges();

    reportDbContext.Add(report);
    reportDbContext.SaveChanges();

    return RedirectToAction("Index");
}

```

Şekil 14 Aracın görevini sonlandırıp rapora yazma işlemi için kullanılan C# kod bloğu

3.2.5. Kayıtlı ürünleri yönetme

Ürünler sayfasında sisteme yeni ürün eklenebilir, var olan ürün düzenlenebilir veya silinebilir. İşletmedeki olası çok sayıda ürünlerin sayfayı uzatıp görüntüyü bozmasının önüne geçmek için PaginatedList sınıfı kullanılarak ürünlere sayfalama sistemi uygulanmıştır. Bazı ürünler ek açıklama gerektirmeyeceği için sisteme boş olarak girilen açıklamalar rahatsız edici bir görüntü oluşturmaması için NULL anahtarını yerine boşluk olarak görüntülenmiştir.

Ürünler

Ürün ekle

İsim	Fiyat	Açıklama	Olaylar
Ekmek	₺5		Düzenle Sil
Ülker Çikolatalı Gofret	₺5		Düzenle Sil
Dondurma	₺5.5		Düzenle Sil
Çokamel	₺4	4'lü Paket	Düzenle Sil

Önceki

Sonraki

Şekil 15 Ürünler sayfası

3.2.5.1. Ürün ekleme

Bir ürün için gerekecek minimal özellikler ele alındığında ayırt edici ve işlevsel olabilmesi adına sadece isim ve fiyat girerek ürün oluşturulabilir. Eğer aynı ürün grubundan farklı paketler oluşturulacaksa açıklama kısmına girilerek karmaşanın önüne geçilebilir.

Oluřtur

Ürün

İsim

Fiyat

Açıklama

Oluřtur

[Listeye Dön](#)

Şekil 16 Ürün oluřturma

3.2.5.2. Ürün düzenleme

Ürün yelpazesini genişletmek isteyen ve mevcut ürünlerini silip baştan oluřturmak istemeyen iřletme sahipleri basit ve hızlı bir şekilde bu ürünleri düzenleyip yeni sistemine uyumlu hâle getirebilir. Raporlar sayfasında ürünler operasyonla anlık olarak ilişkilendirilir dolayısıyla ürünlerde yapılan düzenlemeler geçmiş raporları etkilemezler ayrıca aktif görevdeki araçlar yüklendiğı andaki verileri esas aldığı için ürünlerde yapılan değışiklikler aktif görevdeki araçları etkilemezler.

Düzenle

Ürün

İsim

Ekmek

Fiyat

5

Açıklama

Kaydet

[Listeye Dön](#)

Şekil 17 Ürün düzenleme sayfası

3.2.5.3. Ürün silme

Ürünlere ekleme yapıp işletmeye yeni ürünler eklenebildiği gibi sistemde artık yer almayan ürünler de hızlıca silinebilir. Ürünler arasında isim benzerliği olabileceği için silme sayfasında kullanıcıya ürüne ait açıklama gösterilir. Sistemden silinen ürünler kayıtlı raporlardan silinmezler.

Silmek istediğinden emin misin?

Ürün

İsim: Ekmek

Fiyat: ₺5

Açıklama:

Sil

[Listeye Dön](#)

Şekil 18 Ürün silme sayfası

3.2.6. Rapor sistemi ve z raporu

Sistemde göreve gönderilen ve görevini tamamlayan her araç, taşıdığı ürünler ve gönderildiği adres sisteme rapor olarak kaydedilir. Geriye dönük raporlar arasında arama özelliğini kullanarak spesifik adet ve isim girdileriyle arama yapılabilir veya araç markalarını kullanarak arama yapılabilir. Tabloda mavi renkte gösterilen tüm başlıklar o sütunda gösterilen verilerin sıralanmaya uygun olduğunu belirtirler.

Raporlar sayfasında kullanıcı isterse kendisinin spesifik olarak seçtiği belirli raporları isterse kolayca sayfadaki tüm raporları temizleyebilir. Raporlar sistemi iş yükünün gelir ayağını büyük oranda sistem üzerinde bulunduran kullanıcılar için günlük z raporu özelliği barındırır. Z raporu özelliği raporun oluşturulduğu gün içerisinde gerçekleştirilen tüm operasyonların ürün kalemlerini ve toplam değerlerini listeler.

Raporlar

Tüm Listeye Dön

Z Raporu Çıkart

Ara

Ara

☐ Tümünü Seç

Seçili Raporları Temizle

- Toplam tutar: ₺1950

Seç	Rapor.No	Tarih	Saat	Araç	Adres	Ürünler	Sipariş Tutarı
<input type="checkbox"/>	3	20-05-2021	22:12	Honda Cbf 150	Gölcük	100 adet Ekmek - 100 adet Ülker Çikolatalı Gofret - 100 adet Dondurma - 100 adet Çokomel	₺1950

Önceki

Sonraki

Şekil 19 Raporlar sayfası

```

public async Task<IActionResult> Index(
    string sortOrder,
    string currentFilter,
    string searchString,
    int? pageNumber)
{
    if (pageNumber.HasValue && pageNumber.Value < 1) pageNumber = 1;
    if (sortOrder == null) sortOrder = "id_desc";
    ViewData["CurrentSort"] = sortOrder;
    ViewData["VehicleSortParm"] = sortOrder == "vehicle" ? "vehicle_desc" : "vehicle";
    ViewData["AddressSortParm"] = sortOrder == "address" ? "address_desc" : "address";
    ViewData["PriceSortParm"] = sortOrder == "price" ? "price_desc" : "price";
    ViewData["DateSortParm"] = sortOrder == "Date" ? "date_desc" : "Date";
    ViewData["IdSortParm"] = sortOrder == "Id" ? "id_desc" : "Id";

    if (searchString != null)
    {
        pageNumber = 1;
    }
    else
    {
        searchString = currentFilter;
    }

    ViewData["CurrentFilter"] = searchString;

    var reports = from s in dbContext.Reports.Include(s => s.Vehicle)
                  select s;
    if (!String.IsNullOrEmpty(searchString))
    {
        reports = reports.Where(s => s.Vehicle.Name.Contains(searchString)
                                || s.Date.ToString().Contains(searchString)
                                || s.Items.Contains(searchString)
                                || s.Address.Contains(searchString));
    }
}

```

```

switch (sortOrder)
{
    case "Id":
        reports = reports.OrderBy(s => s.Id);
        break;
    case "id_desc":
        reports = reports.OrderByDescending(s => s.Id);
        break;
    case "price":
        reports = reports.OrderBy(s => s.Total);
        break;
    case "price_desc":
        reports = reports.OrderByDescending(s => s.Total);
        break;
    case "address":
        reports = reports.OrderBy(s => s.Address);
        break;
    case "address_desc":
        reports = reports.OrderByDescending(s => s.Address);
        break;
    case "vehicle_desc":
        reports = reports.OrderByDescending(s => s.Vehicle.Name);
        break;
    case "Date":
        reports = reports.OrderBy(s => s.Date);
        break;
    case "date_desc":
        reports = reports.OrderByDescending(s => s.Date);
        break;
    default:
        reports = reports.OrderBy(s => s.Vehicle.Name);
        break;
}

double total = 0;

foreach (Report report in reports)
{
    total += report.Total;
}

```

Şekil 20 Filtreleme, sıralama işlemini yapan C# kod bloğu

3.2.6.1. Seçme işlemlerinin alt yapısı ve veri gösterimi

Daha önce bölüm 3.2.4.2’de kullanılan mimariye benzer olarak kullanıcının seçimlerine göre “Tümünü Seç” butonunun otomatik olarak aktifleşmesini sağladım. Ayrıca yine aynı mimariyi temel alarak seçili raporların toplam değerini kullanıcıya gösterdim.

```
function selectCheckbox(checked) {  
    var allSelect = document.getElementById("allSelect");  
    if (!checked) {  
        allSelect.checked = false;  
    } else {  
        var checkboxes = $("input[id^='checkbox']");  
        var containsUnchecked = false;  
        for (var i = 0; i < checkboxes.length; i++) {  
            var element = checkboxes[i];  
            if (!element.checked) {  
                containsUnchecked = true;  
                break;  
            }  
        }  
        if (!containsUnchecked) allSelect.checked = true;  
    }  
    calculateReports();  
}
```

Şekil 21 Rapor kutucuğu seçildiğinde çalışan JavaScript kodu

```

function selectAll() {

    var checkbox = document.getElementById("allSelect");

    var checked = checkbox.checked;

    var checkboxes = $("input[id^='checkbox']");

    for (var i = 0; i < checkboxes.length; i++) {

        var element = checkboxes[i];

        element.checked = checked;

    }

    calculateReports();

}

```

Şekil 22 Tümünü işaretle kutucuğu seçildiğinde çalışan JavaScript kodu

```

function calculateReports() {

    var checkboxes = $("input[id^='checkbox']");

    var total = 0.0;

    var found = false;

    for (var i = 0; i < checkboxes.length; i++) {

        var element = checkboxes[i];

        if (element.checked) {

            found = true;

            var price = document.getElementById("price " + element.value);

            var priceFloat = parseFloat(price.innerHTML.replace('₺', '').replace(',', '.'));

            total += priceFloat;

        }

    }

    var selected = document.getElementById("selected");

    if (found) {

        selected.style.display = "";

        selected.innerHTML = "Seçili Raporların Toplamı ₺" + total;

    } else selected.style.display = "none";

}

```

Şekil 23 Seçilen raporların toplam fiyatını gösteren JavaScript kodu

3.2.6.2. Z raporu

İşletmelerin lojistik operasyonları maddi gelirlerin tamamını karşılıyor olabilir. Bu tarz durumlarda küçük çaplı işletmelerde yaygın olarak kullanılan z raporu özelliği oldukça önem kazanmaktadır. Kullanıcının gün içinde yapılan tüm satışların hızlıca içeriklerini ve toplam değerlerini görebileceği, gün içinde yapılan toplam operasyon sayısını ve detaylı incelemek isterse raporu ayırt edebilmesi için rapora ait benzersiz kimlik numarasını barındırır. Küçük çaplı işletmeler yerel olarak saklamayı tercih edebilecekleri için z raporları yazdırılabilir olarak tasarlanmıştır. Eğer kullanıcı “İndir” butonuna tıklarsa sayfadaki işlevsel objeler otomatik olarak gizlenir ve yalnızca rapor kullanıcıya iletilir.

Paket Servis Araç Takip

20.05.2021

Z Raporu

S. No	Rapor No	Ürünler	Sipariş Tutarı
1	3	100 adet Ekmek - 100 adet Ülker Çikolatalı Gofret - 100 adet Dondurma - 100 adet Çokomel	₺1950

Toplam: ₺1950

İndir

Şekil 24 Z raporu

3.2.6.2.1. Z raporunu popup olarak açma ve yazdırma sistemleri

Z raporu, kendisine erişilen sayfadan bağımsız bir pencerede açılarak kullanıcının deneyimini etkilemeden raporu sunmayı hedefler. Z raporunda bulunan işlevsel objeler “non-printable” sınıfıyla işaretlenmişlerdir. Bu sınıf sayfanın yazdırılması veya indirilmesi durumunda işlevsel butonların gizlenmesini sağlayarak tam bir rapor görünümü sunar.

```
function showPopUp() {  
    window.open('/Report/ZReport', "PopupWindow",  
        'width=800px,height=800px,top=150,left=250');  
}
```

Şekil 25 Z Raporunu popup olarak açmak için kullanılan JavaScript kodu

3.2.7. Kullanıcı paneli ve profil yönetimi

Kullanıcı bu kısımda kendi hesabına ait telefon numarası, email ve şifre bilgilerini değiştirebilir, sisteme kayıtlı kişisel verilerini indirebilir veya sistemden hesabıyla birlikte silebilir. Ayrıca sisteme yeni kullanıcı ekleyebilir.

Hesabını yönet

Hesap ayarlarını değiştir

Profil	Profil
Email	Kullanıcı adı
Şifre	<input type="text" value="test@gmail.com"/>
Yeni Kullanıcı Oluştur	Telefon numarası
Kişisel veri	<input type="text"/>
	Kaydet

Şekil 26 Yönetici paneli

3.2.7.1. Email yönetimi

Kullanıcı bu sayfada hesabına tanımlanmış olan maili kendisine verilmiş olan emaili değiştirebilir. Fakat bu işlem güvenlik gerekçesiyle eski maili doğrulamadan gerçekleştirilemez.

Hesabını yönet

Hesap ayarlarını değiştir

[Profil](#)[Email](#)[Şifre](#)[Yeni Kullanıcı Oluştur](#)[Kişisel veri](#)

Emaili Yönet

Email

test@gmail.com

[Email onaylaması gönder](#)

Yeni email

test@gmail.com

Emaili değiştir

Şekil 27 Email yönetim sayfası

3.2.7.2. Şifre yönetimi

Kullanıcıya verilmiş olan şifre güvenlik sıkıntıları nedeniyle açığa çıkmış ise veya kullanıcı herhangi bir sebepten şifresini değiştirmek isterse bu sayfadan mevcut şifresini ve yeni şifresini girerek şifresini güncelleyebilir. Şifreler sistemde MVC 3.1 standartı olan PBKDF 2 ile şifrelenerek saklandığı için kullanıcıdan istenen eski şifre hiçbir güvenlik açığına sebep olmadan kontrol edilir.

Hesabını yönet

Hesap ayarlarını değiştir

[Profil](#)[Email](#)[Şifre](#)[Yeni Kullanıcı Oluştur](#)[Kişisel veri](#)

Şifre değiştir

Mevcut şifre

Yeni şifre

Yeni şifreyi onayla

Şifreyi güncelle

Şekil 28 Şifre yönetimi

3.2.7.3. Yeni kullanıcı oluşturma

Yönetici, herhangi bir çalışanına mevcut yönetici hesabını vermek yerine harici bir kullanıcı oluşturarak atama yapabilir. Bu sayfada yeni kullanıcıya verilecek email hesabı ve şifre alanları doldurularak yeni hesap oluşturulur. Yeni kullanıcıya tanımlanan mail adresi kullanıcı ekleme esnasında mevcut olmayabilir. Yönetici tarafından tanımlandığı için ek bir doğrulamaya gerek görülmez.

Hesabını yönet

Hesap ayarlarını değiştir

Profil	Yeni Kullanıcı Oluştur
Email	Email
Şifre	<input type="text"/>
Yeni Kullanıcı Oluştur	Şifre
Kişisel veri	<input type="text"/>
	Oluştur

Şekil 29 Yeni kullanıcı oluşturma

3.2.7.4. Kişisel veri yönetimi

Sistemin kullanıcı hakkında topladığı verileri kullanıcılar bu sayfadan talep edebilirler. Kullanıcı artık sistemde yer almıyorsa veya yönetici kullanıcının sistemde artık yer almasını istemiyorsa hesabına girerek yine bu sayfadan hesabını silebilir. Bu veriler kullanıcının sistemdeki benzersiz kimliği, kullanıcı adı, mail adresi, telefon numarası ayrıca telefon ve mail adreslerinin doğrulanma durumlarını belirten bir değer içerir.

Hesabını yönet

Hesap ayarlarını deęiřtir

Profil	<h3>Kiřisel Veri</h3>
Email	Hesabın bize verdięin kiřisel veriyi ierir. Bu sayfa bu veriyi indirmeni veya silmeni saęlar.
řifre	Bu veriyi silmek hesabını kalıcı olarak siler ve geri alınamaz.
Yeni Kullanıcı Oluřtur	
Kiřisel veri	İndir
	Sil

řekil 30 Kiřisel veri sayfası

4. SONUÇLAR VE ÖNERİLER

Ürettiğimiz projede küçük ölçekli yerel hizmet sağlayıcılarının lojistik operasyonlarını yönetmesini amaçladık. Bu yönetim sürecinde maliyetlerin minimize edilmesi ve verimin artırılması için çoklu platform desteğini ön planda tuttuk. Sistemin fizibilitesi değerlendirildiğinde elemanların günümüzde bağımsız platformları tercih etme olasılıklarının oldukça yüksek olmasından dolayı sistemin çoklu platform uyumu öne çıkmaktadır. Bu ve benzeri platformlarda hayati işlevlerin dışında kalan ek özelliklerden ziyade önceliğin çoklu platform konusuna verilmesi gerektiği sonucuna varılabilir. Proje geliştirilirken tercih edilen ASP.NET Core MVC teknolojisi açık kaynaklı ve küçük ölçekli işletmeleri hedefleyen web uygulamaları geliştirmek için sunduğu temel yapı ve mimarisi bakımından oldukça uygun olmasına rağmen büyük projelerin de mimarisinin sağlam temelleri sayesinde hayata geçirilmesine olanak sağlar.

Geliştirdiğimiz uygulama küçük işletmeleri öncelediği için projeye bir rol sistemi eklenmedi. Ölçeği büyütmek isteyen kişilerin erişim ve düzenleme noktalarında yetki tabanlı rol sistemi eklemesi işletmelerinin güvenliği için de azami özen gösterilmesi gereken bir noktadır. Günümüz regülasyonlarına uyum sağlayan ve azami derecede faydalanan bir sistem olması açısından geliştirmek isteyen kişiler elektronik fatura sistemi entegre edebilirler.

KAYNAKLAR

- [1] https://tr.wikipedia.org/wiki/Microsoft_Visual_Studio
- [2] https://www.w3schools.com/cs/cs_intro.asp
- [3] <https://docs.microsoft.com/tr-tr/dotnet/csharp/tour-of-csharp/>
- [4] https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_overview.htm
- [5] https://www.tutorialspoint.com/entity_framework/entity_framework_overview.htm
- [6] https://www.tutorialspoint.com/asp.net_mvc/asp.net_mvc_razor.htm
- [7] <https://www.tutorialspoint.com/bootstrap/index.htm>

EKLER

Ek-1

Paket Servis Araç Takip

[Ana Sayfa](#) [Araçlar](#) [Ürünler](#) [Raporlar](#) [Merhaba test@gmail.com!](#) [Çıkış Yap](#)

Hoş geldiniz

Paket servis araç takip by [Yasin Subaşı](#).

© 2021 - Paket Servis Araç Takip

Ek-2

Paket Servis Araç Takip

Giriş Yap



E-Posta

Şifre

☐ Beni hatırla

Giriş Yap

© 2021 - Paket Servis Araç Takip

Ek-3

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Araçlar

Araç ekle

Araç No	İsim	Ürünler	Adres	Olaylar
6	Honda Cbf 150			Düzenle Sil Yükleme Yap Görevi Sonlandır
7	Haojue 135 Da			Düzenle Sil Yükleme Yap Görevi Sonlandır
8	Bajaj Pulsar 200 Ns			Düzenle Sil Yükleme Yap Görevi Sonlandır
9	Tvs Apache RTR 150			Düzenle Sil Yükleme Yap Görevi Sonlandır
10	Yamaha Ybr 125			Düzenle Sil Yükleme Yap Görevi Sonlandır

Ek-4

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Oluştur

Araç

İsim

Oluştur

[Listeye Dön](#)

© 2021 - Paket Servis Araç Takip

Ek-5

Paket Servis Araç Takip

[Ana Sayfa](#) [Araçlar](#) [Ürünler](#) [Raporlar](#) [Merhaba test@gmail.com!](#) [Çıkış Yap](#)

Düzenle

Araç

İsim

Honda Cbf 150

Kaydet

[Listeye Dön](#)

© 2021 - Paket Servis Araç Takip

Ek-6

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Silmek istediğinden emin misin?

Araç

Araç No: 6
İsim: Honda Cbf 150

Sil

[Listeye Dön](#)

© 2021 - Paket Servis Araç Takip

Ek-7

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Honda Cbf 150 İsimli Araca Yükleme Yap

Yükleme yap

Adres:

Yüklensin mi?	Adet	İsim	Fiyat	Açıklama
<input type="checkbox"/>	1	Ekmek	₺5	
<input type="checkbox"/>	1	Ülker Çikolatalı Gofret	₺5	
<input type="checkbox"/>	1	Dondurma	₺5,5	
<input type="checkbox"/>	1	Çokomel	₺4	4'lü Paket
Yüklensin mi?	Adet	İsim	Fiyat	Açıklama

Listeye dön

© 2021 - Paket Servis Araç Takip

Ek-8

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Raporlar

Tüm Listeye Dön

Z Raporu Çıkart

Ara

Ara

☐ Tümünü Seç

• Toplam tutar: ₺1980

Seçili Raporları Temizle

Seç	Rapor No	Tarih	Saat	Araç	Adres	Ürünler	Sipariş Tutarı
<input type="checkbox"/>	10	24-05-2021	17:30	Honda Cbf 150	İzmit	5 adet Ekmek - 1 adet Ülker Çikolatalı Gofret	₺30
<input type="checkbox"/>	3	20-05-2021	22:12	Honda Cbf 150	Gölcük	100 adet Ekmek - 100 adet Ülker Çikolatalı Gofret - 100 adet Dondurma - 100 adet Çökmel	₺1950

Önceki

Sonraki

© 2021 - Paket Servis Araç Takip

Ek-9

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Hesabını yönet

Hesap ayarlarını değiştir

Profil

Email

Şifre

[Yeni Kullanıcı Oluştur](#)

[Kişisel veri](#)

Profil

Kullanıcı adı

test@gmail.com

Telefon numarası

Kaydet

© 2021 - Paket Servis Araç Takip

Ek-10

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Hesabını yönet

Hesap ayarlarını değiştir

Profil

Email

Şifre

Yeni Kullanıcı Oluştur

Kişisel veri

Emaili Yönet

Email

test@gmail.com

[Email onaylaması gönder](#)

Yeni email

test@gmail.com

Emaili değiştir

© 2021 - Paket Servis Araç Takip

Ek-11

Paket Servis Araç Takip

Ana Sayfa Araçlar Ürünler Raporlar Merhaba test@gmail.com! Çıkış Yap

Hesabını yönet

Hesap ayarlarını değiştir

Profil

Email

Şifre

Yeni Kullanıcı Oluştur

Kişisel Veri

Şifre değiştir

Mevcut şifre

Yeni şifre

Yeni şifreyi onayla

Şifreyi güncelle

Ek-12

Paket Servis Araç Takip

[Ana Sayfa](#) [Araçlar](#) [Ürünler](#) [Raporlar](#) [Merhaba test@gmail.com!](#) [Çıkış Yap](#)

Hesabını yönet

Hesap ayarlarını değiştir

[Profil](#)

[Email](#)

[Şifre](#)

[Yeni Kullanıcı Oluştur](#)

[Kişisel veri](#)

Yeni Kullanıcı Oluştur

Email

Şifre

[Oluştur](#)

Ek-13

Hesabını yönet

Hesap ayarlarını değiştir

[Profil](#)

[Email](#)

[Şifre](#)

[Yeni Kullanıcı Oluştur](#)

[Kişisel veri](#)

Kişisel Veri

Hesabın bize verdiğin kişisel veriyi içerir. Bu sayfa bu veriyi indirmeni veya silmeni sağlar.

Bu veriyi silmek hesabını kalıcı olarak siler ve geri alınamaz.

[İndir](#)

[Sil](#)

ÖZGEÇMİŞ

2000 yılında İstanbul'un Zeytinburnu ilçesinde dünyaya geldi. Daha sonradan Kocaeli'nin Gölcük ilçesinde orta ve lise eğitimini tamamladı. 2019 yılında Kocaeli Üniversitesi Köseköy Yüksekokulu Bilgisayar Programcılığını kazandı. Eğitimine hâlen burada devam etmektedir. 2016 yılında Java ile tanıştığundan beri yazılıma ilgi duymaktadır ve uluslararası bir platformda Java ile geliştirdiği içerikleri satmaktadır.