Telefon Bayi Stok Takip ve Satış Otomasyonu

Ayşegül TÜTÜNCÜ, Esma GELMEZ, Binnur ÖZCAN May 15, 2023

Abstract

Bu projenin amacı, bir telefon bayisinin stok takibi ve satış yönetimini kolaylaştırmaktır. Bu amaç doğrultusunda, bir veritabanı şeması oluşturulmuş ve bu şema kullanılarak bir MVC uygulaması geliştirilmiştir. Uygulama, kullanıcılara ürünleri görüntüleme, stok takibi yapma, satış yapma ve kampanyaları yönetme gibi işlevler sunmaktadır.

1 Giriş

Bu rapor, bir telefon satış ve stok takip otomasyonu projesinin IEEE formatında sunumunu içermektedir. Bu projenin amacı, bir telefon bayisinin stok takibi ve satış yönetimini kolaylaştırmaktır. Bu amaç doğrultusunda, bir veritabanı şeması oluşturulmuş ve bu şema kullanılarak bir MVC uygulaması geliştirilmiştir.

2 Proje Tanımı

Bu projenin amacı, bir telefon bayisinin stok takibi ve satış yönetimini kolaylaştırmaktır. Bu amaç doğrultusunda, bir veritabanı şeması oluşturulmuş ve bu şema kullanılarak bir MVC uygulaması geliştirilmiştir. Uygulama, kullanıcılara ürünleri görüntüleme, stok takibi yapma, satış yapma ve kampanyaları yönetme gibi işlevler sunmaktadır.

3 Kullanılan Teknolojiler

Bu projede, C# programlama dili kullanılmıştır. Veritabanı yönetimi için SQL Server kullanılmıştır. Veritabanı şeması, müşteriler, ürünler, kampanyalar, kategoriler, satışlar ve satış detayları gibi farklı tabloları içermektedir. Bu tablolar arasındaki ilişkiler, FOREIGN KEY kavramı kullanılarak tanımlanmıştır. Ayrıca, her tabloda birincil anahtar alanları identity olarak tanımlanmıştır.

4 Veritabanı Tasarımı

Veritabanı tasarımı, müşteriler, ürünler, kampanyalar, kategoriler, satışlar ve satış detayları gibi farklı tabloları içermektedir. Bu tablolar arasındaki ilişkiler, FOREIGN KEY kavramı kullanılarak tanımlanmıştır. Ayrıca, her tabloda birincil anahtar alanları identity olarak tanımlanmıştır. Veritabanı müşteriler, ürünler, kampanyalar, kategoriler, satışlar ve satış detayları gibi farklı tabloları içermektedir. Müşteriler tablosu, müşteri bilgilerini içermektedir. Bu tablo, satışlar tablosu ile birincil anahtar (primary key) alanı olan "musterilD" sütunu aracılığıyla ilişkilidir. Satışlar tablosu arasında bir "birçok" ilişkisi vardır. Ürünler tablosu, ürün bilgilerini içermektedir. Bu tablo, satış detayları tablosu ile birincil anahtar alanı olan "urunID" sütunu aracılığıyla ilişkilidir. Satış detayları tablosu ile satış işlemi için bir veya daha fazla ürün satın alınabilir. Bu nedenle, ürünler tablosu ile satış detayları tablosu arasında bir "birçok" ilişkisi vardır. Kampanyalar tablosu, kampanya bilgilerini içermektedir. Bu tablo, satışlar tablosu ile birincil anahtar alanı olan "kampanyaID" sütunu aracılığıyla ilişkilidir. Satışlar tablosunda, her bir satış işlemi için bir kampanya uygulanabilir. Bu nedenle, kampanyalar tablosu ile satışlar tablosu arasında bir "birçok" ilişkisi vardır. Kategoriler tablosu, ürün kategorilerini

içermektedir. Bu tablo, ürünler tablosu ile birincil anahtar alanı olan "kategoriID" sütunu aracılığıyla ilişkilidir. Ürünler tablosunda, her bir ürün bir kategoriye atanmıştır. Bu nedenle, kategoriler tablosu ile ürünler tablosu arasında bir "birçok" ilişkisi vardır. Satışlar tablosu, satış işlemlerini içermektedir. Bu tablo, satış detayları tablosu ile birincil anahtar alanı olan "satisID" sütunu aracılığıyla ilişkilidir. Satış detayları tablosunda, her bir satış işlemi için bir veya daha fazla satış detayı kaydı bulunmaktadır. Bu nedenle, satışlar tablosu ile satış detayların içermektedir. Bu tablo, ürünler tablosu ile birincil anahtar alanı olan "urunID" sütunu aracılığıyla ilişkilidir. Ürünler tablosunda, her bir ürün için bir veya daha fazla satış detayı kaydı bulunmaktadır. Bu nedenle, ürünler tablosu ile satış detayları tablosu arasında bir "birçok" ilişkisi vardır. Bu şekilde, veritabanı şemasındaki tablolar arasındaki ilişkiler, birbirleriyle bağlantılı verilerin doğru bir şekilde depolanmasını ve yönetilmesini sağlar. Bu sayede, bir telefon satış ve stok takip otomasyonu projesinde, müşteri, ürün, kampanya ve satış bilgileri gibi farklı verilerin doğru bir şekilde takip edilmesi ve yönetilmesi mümkün olur.

5 MVC Uygulaması

MVC uygulaması, Model-View-Controller mimarisine dayanmaktadır. Model, veritabanı şemasını temsil eder ve verilerin depolanmasını ve yönetilmesini sağlar. View, kullanıcı arayüzünü temsil eder ve kullanıcının uygulamayla etkileşimini sağlar. Controller, kullanıcının girdilerini alır ve Model ve View arasındaki etkileşimi yönetir. Uygulama, kullanıcılara ürünleri görüntüleme, stok takibi yapma, satış yapma ve kampanyaları yönetme gibi işlevler sunmaktadır. Telefon stok takip ve satış otomasyonu, ASP.NET MVC kullanılarak geliştirilmiştir. Aşağıda, kod parçalarının kullanımı hakkında daha ayrıntılı bilgi verilmiştir. Yeni Müşteri Ekleme Sayfası Yeni müşteri ekleme sayfası, bir ASP.NET MVC View sayfasıdır. Bu sayfa, müşteri adı, soyadı ve telefon numarası gibi bilgileri girmek için bir form içerir. Form, post metodu ile gönderilir ve yeni müşteri eklemek için kullanılır. Ürün Listeleme Sayfası Urün listeleme sayfası, bir ASP.NET MVC View sayfasıdır. Bu sayfa, URUNLER modelindeki ürünleri listeleyen bir tablo içerir. Tablo, ürün ID'si, kategori, marka, model, açıklama, fiyat ve stok durumu gibi bilgileri gösterir. Her bir ürün icin, silme ve güncelleme bağlantıları sağlanır. Ayrıca, yeni bir ürün eklemek için bir bağlantı da sağlanır. Ürün Ekleme Sayfası Ürün ekleme sayfası, bir ASP.NET MVC View sayfasıdır. Bu sayfa, yeni bir ürün eklemek için bir form içerir. Form, post metodu ile gönderilir ve ürünün kategori, marka, model, açıklama, fiyat ve stok durumu gibi bilgilerini içerir. Ürün Güncelleme Sayfası Ürün güncelleme sayfası, bir ASP.NET MVC View sayfasıdır. Bu sayfa, mevcut bir ürünü güncellemek için bir form içerir. Form, post metodu ile gönderilir ve ürünün kategori, marka, model, açıklama, fiyat ve stok durumu gibi bilgilerini içerir. Ürün Silme Sayfası Ürün silme sayfası, bir ASP.NET MVC View sayfasıdır. Bu sayfa, mevcut bir ürünü silmek için bir onay formu içerir. Form, post metodu ile gönderilir ve ürünün silinmesini onaylamak için kullanılır. Veritabanı Yapısı Veritabanı yapısı, Entity Framework kullanılarak oluşturulmuştur. Veritabanı, müşteri ve ürün bilgilerini içeren iki tablo içerir. Müşteri tablosu, müşteri adı, soyadı ve telefon numarası gibi bilgileri içerir. Urün tablosu, kategori, marka, model, açıklama, fiyat ve stok durumu gibi bilgileri içerir. Controller Sınıfları Controller sınıfları, ASP.NET MVC uygulamasının iş mantığını yönetir. Her bir sayfa için bir controller sınıfı oluşturulur. Controller sınıfları, veritabanı işlemlerini gerçekleştirir ve View sayfalarına veri sağlar. Sonuç olarak, yukarıda verilen kod parçaları, bir telefon stok takip ve satış otomasyonu için kullanılmıştır. Bu kod parçaları, ASP.NET MVC kullanılarak geliştirilmiş ve müşteri ve ürün bilgilerini içeren bir veritabanı kullanmaktadır. Bu otomasyon, müşteri ve ürün bilgilerini takip etmek, yeni müsteriler ve ürünler eklemek, mevcut ürünleri güncellemek ve silmek icin kullanılabilir.

6 SQL Server'da Index, Trigger ve View Kullanımı

6.1 SQL Server'da Index Kullanımı

Index, veritabanı sorgularının daha hızlı çalışmasını sağlamak için kullanılan bir yapıdır. Index, bir tablodaki verileri belirli bir sıraya göre düzenler ve burade sorguların daha hızlı çalışmasını sağlar. Index, bir veya daha fazla sütuna uygulanabilir.

Index oluşturmak için aşağıdaki SQL sorgusu kullanılabilir:

CREATE INDEX index_adi ON tablo_adi(sutun_adi)

Örneğin, "urunler" tablosundaki "urunAdi" sütununa index oluşturmak için aşağıdaki SQL sorgusu kullanılabilir:

CREATE INDEX idx_urunler_urunAdi ON urunler(urunAdi)

Index, veritabanı performansını artırmak için önemli bir araçtır. Ancak, index oluşturmak, veritabanı boyutunu artırabilir ve veri ekleme, güncelleme ve silme işlemlerini yavaşlatabilir. Bu nedenle, index oluştururken dikkatli olunmalı ve gereksiz index'lerden kaçınılmalıdır.

6.2 SQL Server'da Trigger Kullanımı

Trigger, bir veritabanı işlemi gerçekleştiğinde otomatik olarak çalışan bir işlevdir. Trigger, INSERT, UPDATE veya DELETE işlemleri gibi belirli bir veritabanı işlemi gerçekleştiğinde tetiklenebilir.

Trigger oluşturmak için aşağıdaki SQL sorgusu kullanılabilir:

CREATE TRIGGER trigger_adi ON tablo_adi AFTER INSERT, UPDATE, DELETE AS BEGIN END

Örneğin, "urunler" tablosuna yeni bir ürün eklendiğinde, "urunler_log" tablosuna otomatik olarak bir kayıt eklemek için aşağıdaki SQL sorgusu kullanılabilir:

CREATE TRIGGER tr_urunler_insert ON urunler AFTER INSERT AS BEGIN INSERT INTO urunler_log(urunAdi, fiyat, stok) SELECT urunAdi, fiyat, stok FROM inserted END

Bu trigger, "urunler" tablosuna yeni bir ürün eklendiğinde, "urunler_log" tablosuna otomatik olarak bir kayıt ekler.

6.3 SQL Server'da View Kullanımı

View, bir veya daha fazla tablodan veri alarak, bu verileri bir sanal tablo olarak sunan bir yapıdır. View, veritabanı sorgularının daha kolay ve anlaşılır hale gelmesini sağlar.

View oluşturmak için aşağıdaki SQL sorgusu kullanılabilir:

CREATE VIEW_adi AS SELECT sutun_adi1, sutun_adi2, ... FROM tablo_adi WHERE kosul

Örneğin, "musteriler" tablosundaki "ad" ve "soyad" sütunlarını birleştirerek "ad_soyad" adlı bir view oluşturmak için aşağıdaki SQL sorgusu kullanılabilir:

CREATE VIEW vw_musteriler_ad_soyad AS SELECT ad + ' ' + soyad AS ad_soyad FROM musteriler

Bu view, "musteriler" tablosundaki "ad" ve "soyad" sütunlarını birleştirerek "ad_soyad" adlı bir sanal tablo oluşturur. Bu view, veritabanı sorgularının daha kolay ve anlaşılır hale gelmesini sağlar.

7 Sonuç

Projenin geliştirme süreci, aşağıdaki adımları içermiştir: Veritabanı tasarımı; Veritabanı tasarımı, SQL Server Management Studio kullanılarak yapılmıştır. Veritabanı, ürünler, müşteriler, satışlar ve stok seviyeleri gibi verileri içermektedir. SQL Server'da create ile tabloları oluşturup insert ile veri eklemesi yapılmıştır. SQL Server'da index, trigger ve view kullanımı, veritabanı yönetimi ve performansı açısından önemlidir. İndex, sorguların daha hızlı çalışmasını sağlar, trigger, veritabanı işlemlerinin otomatik olarak gerçekleştirilmesini sağlar ve view, veritabanı sorgularının daha kolay ve anlaşılır hale gelmesini sağlar. Ancak, bu yapıların kullanımı dikkatli bir şekilde yapılmıştır. Bu mimari, uygulamanın farklı katmanlarını ayrı ayrı ele alarak, daha modüler ve esnek bir yapı oluşturulmasına olanak tanır. Projenin tasarımı aşağıdaki katmanlardan oluşmaktadır.

Model katmanı: Model katmanı, veritabanı işlemleri ve veri işleme işlemleri için oluşturulmuştur. Bu katmanda, Entity Framework kullanılarak veritabanı işlemleri gerçekleştirilmiştir. View katmanı: View katmanı, kullanıcı arayüzü için oluşturulmuştur. Bu katmanda, Razor sözdizimi kullanılarak HTML kodları ve C# kodları bir arada kullanılmıştır. Controller katmanı: Controller katmanı, kullanıcının girdilerini işleyen ve model ve view arasındaki iletişimi sağlayan işlemler için oluşturulmuştur. Bu katmanda, C# kodları kullanılarak işlemler gerçekleştirilmiştir. Test: Projenin test süreci, Visual Studio kullanılarak yapılmıştır. Testler, proje gereksinimlerine uygunluğu kontrol etmek için yapılmıştır.

8 Sonuçlar ve Tartışma

Bu proje, bir telefon bayisinin stok takibi ve satış yönetimini kolaylaştırmak için bir MVC uygulaması geliştirmeyi amaçlamaktadır. Uygulama, C# programlama dili kullanılarak geliştirilmiş ve SQL Server'da index, trigger ve view kullanmaktadır. Uygulama, kullanıcılara ürünleri görüntüleme, stok takibi yapma, satış yapma ve kampanyaları yönetme gibi işlevler sunmaktadır.

9 KAYNAKÇA

https://openai.com/

https://www.youtube.com/playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlist?list=PLi1BmHvgBkxI4uBS5vjfhqhAUWPnRL73-playlistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphaylistphay

https://www.btkakademi.gov.tr/portal/course/uygulamalarla-sql-ogreniyorum-8249

https://medium.com/baakademi/normali

https://www.youtube.com/watch?v=gn9lekIceyclist=PLzJP4sAivSii2VDD49eArBnR-D-04MR2F

https://github.com/Burak006/Telefon-Satis-Otomasyonu