

RESTORAN POS SİSTEMİ PROJE TANIM DOKÜMANI

Mehmethan Doru

İÇİNDEKİLER

İÇİNDEKİLER	2
Giriş	5
Teknoloji Yığını (Tech Stack) Bilgilendirmesi	5
Frontend: React	5
Backend: Go (Golang)	5
Diğer Araçlar:	5
Uygulama Özellikleri (Features)	6
Kullanıcı Yönetimi	6
Kullanıcı Kaydı ve Girişi:	6
Profil Yönetimi:	6
Kategori Yönetimi	6
Kategori Oluşturma:	6
Kategori Güncelleme ve Silme:	6
Ürün Yönetimi	6
Ürün Ekleme:	6
Ürün Güncelleme ve Silme:	6
Ürün Listeleme:	6
Sepet İşlemleri	7
Sepete Ürün Ekleme:	7
Sepetten Ürün Çıkarma ve Güncelleme:	7
Sepeti Boşaltma:	7
Sipariş ve Fatura Yönetimi	7
Sipariş Oluşturma:	7
Faturalandırma:	7
Veri Görselleştirme	7
İstatistiksel Analiz:	7
Grafik ve Tablo Gösterimi:	7
Temel API'lar	7
Kullanıcı API'si	7
Kullanıcı Kaydı:	7
Kullanıcı Girişi:	7
Kullanıcı Bilgileri:	8
Kategori API'si	8

Kategori Listeleme:	8
Kategori Oluřturma:	8
Kategori Gncelleme:	8
Kategori Silme:	8
rn API'si	8
rn Listeleme:	8
rn Ekleme:	8
rn Gncelleme:	8
rn Silme:	8
Sepet API'si	8
Sepete rn Ekleme:	8
Sepetten rn ıkarma:	8
Sepetteki rn Gncelleme:	8
Sepeti Bořaltma:	9
Sipariř API'si	9
Sipariř Oluřturma:	9
Faturalandırma ve Yazdırma:	9
Veri Grselleřtirme API'si	9
İstatistiksel Analiz:	9
Frontend (n Yz) İerięi	9
Giriř ve Kayıt Sayfaları	9
Ana Sayfa	9
Kategori Ynetimi Sayfası	9
rn Ynetimi Sayfası	9
Sepet Sayfası	10
Sipariř ve Fatura Sayfası	10
Veri Grselleřtirme Sayfası	10
Backend (Arka Taraf) İerięi	10
Kullanıcı Ynetimi	10
Kategori ve rn Ynetimi	10
Sepet ve Sipariř Ynetimi	10
Veritabanı Ynetimi	10
Veri Grselleřtirme	10
Ana Mimari ve Akıř Diyagramları	11
Ana Mimari	11
Kullanıcı Akıř Diyagramları	11

Veri Akış Diyagramları	12
Test Senaryoları	12
Birim Testleri (Unit Tests)	12
Kullanıcı Kaydı ve Girişi:.....	12
Profil Yönetimi:	12
Kategori Yönetimi:.....	13
Ürün Yönetimi:	13
Sepet Yönetimi:.....	13
Fatura Oluşturma:.....	13
Entegrasyon Testleri (Integration Tests)	13
Kullanıcı Kaydı ve Girişi:.....	13
Profil Yönetimi:	13
Kategori Yönetimi:.....	13
Ürün Yönetimi:	13
Sepet ve Sipariş Yönetimi:	13
Veri Görselleştirme:.....	14
Özet	14
Projenin Amacı ve Kapsamı	14
Teknoloji Yığını.....	14
Ana Özellikler	14
Test Senaryoları.....	14

Giriş

Bu analiz dokümanı, React, Go ve MongoDB kullanılarak geliştirilecek bir POS (Point of Sale) uygulamasının teknik özelliklerini ve geliştirme sürecini detaylandırmaktadır. Proje, kullanıcı kimlik doğrulama, kategori yönetimi, ürün işlemleri, sepet ve sipariş yönetimi gibi çekirdek fonksiyonları içermekte olup, verilerin görselleştirilmesi ile işletim verimliliğini artırmayı hedeflemektedir. Bu doküman, uygulamanın mimari yapısını, API uç noktalarını ve test senaryolarını kapsayarak geliştirme sürecine rehberlik edecektir.

Teknoloji Yığını (Tech Stack) Bilgilendirmesi

Bu proje için tercih edilen teknoloji yığını, performans, ölçeklenebilirlik ve geliştirilebilirlik kriterleri doğrultusunda seçilmiştir:

Frontend: React

Durum Yönetimi: Redux-Toolkit, uygulama durumunun merkezi ve tutarlı bir şekilde yönetilmesi için.

UI Kütüphanesi: Ant Desing, Tailwind CSS ve/veya NextUI, kullanıcı arayüzü bileşenlerinin hızlı ve estetik oluşturulması için.

API İletişimi: Axios veya Fetch API, backend ile veri alışverişi sağlamak için.

Backend: Go (Golang)

Framework(?): Gin, yüksek performanslı HTTP sunucusu ve RESTful API geliştirilmesi için.

Veritabanı: MongoDB, esnek veri modelleme ve hızlı veri erişimi için.

Kimlik Doğrulama: JWT (JSON Web Tokens), güvenli kullanıcı kimlik doğrulama ve oturum yönetimi için.

Diğer Araçlar:

Docker: Uygulamanın farklı ortamlar arasında taşınabilirliğini sağlamak ve çalıştırmak için.

Git: Sürüm kontrolü ve ekip iş birliği yönetimi için.

CI/CD: Jenkins, otomatik entegrasyon ve dağıtım süreçlerinin yönetilmesi için.

Veri Görselleştirme: Ant Design Charts, verilerin etkili ve kullanıcı dostu bir şekilde görselleştirilmesi için.

Uygulama Özellikleri (Features)

POS uygulaması, perakende işletmelerin ihtiyaçlarına yönelik kapsamlı özellikler sunarak kullanıcıların işlemlerini kolaylaştırmayı hedeflemektedir. İşte uygulamanın temel özellikleri:

Kullanıcı Yönetimi

Kullanıcı Kaydı ve Girişi:

Yeni kullanıcıların sisteme kaydolmasını sağlamak ve mevcut kullanıcıların giriş yapabilmesi için kimlik doğrulama mekanizması.

Şifrelerin güvenli bir şekilde saklanması ve kimlik doğrulamanın JWT ile yapılması.

Profil Yönetimi:

Kullanıcıların profil bilgilerini görüntüleyip güncelleyebileceği bir arayüz.

Kategori Yönetimi

Kategori Oluşturma:

Yeni kategoriler ekleyebilme özelliği. Ürünleri sınıflandırmak için kategori oluşturma imkanı.

Kategori Güncelleme ve Silme:

Mevcut kategorilerin düzenlenmesi veya ihtiyacı olmayanların silinmesi.

Ürün Yönetimi

Ürün Ekleme:

Sisteme yeni ürünler ekleyebilme özelliği. Ürün adı, fiyatı, açıklaması, ve stoğu gibi bilgilerin kaydedilmesi.

Ürün Güncelleme ve Silme:

Mevcut ürün bilgilerinin güncellenmesi veya gereksiz ürünlerin sistemden kaldırılması.

Ürün Listeleme:

Kullanıcıların tüm ürünleri görüntüleyebileceği bir listeleme arayüzü.

Sepet İşlemleri

Sepete Ürün Ekleme:

Kullanıcıların ürünleri alışveriş sepetine eklemesini sağlayan bir sistem.

Sepetten Ürün Çıkarma ve Güncelleme:

Sepetteki ürünlerin miktarını güncelleme veya istenmeyen ürünleri sepetten çıkarma imkanı.

Sepeti Boşaltma:

Tüm sepetin tek bir işlemle boşaltılması.

Sipariş ve Fatura Yönetimi

Sipariş Oluşturma:

Kullanıcıların sepetlerindeki ürünler ile sipariş oluşturması.

Faturalandırma:

Oluşturulan siparişlerin faturalandırılması, faturaların PDF formatında indirilebilmesi ve yazdırılabilmesi.

Veri Görselleştirme

İstatistiksel Analiz:

Satışlar, stok durumu ve kullanıcı davranışlarına ilişkin istatistiksel verilerin analizi.

Grafik ve Tablo Gösterimi:

Ant Design Charts kullanılarak verilerin grafiksel ve tablo şeklinde görselleştirilmesi.

Temel API'lar

POS uygulaması için geliştirilmiş API uç noktaları:

Kullanıcı API'si

Kullanıcı Kaydı:

Endpoint: POST /api/register , Yeni kullanıcı kaydı oluşturur.

Kullanıcı Girişi:

Endpoint: POST /api/login , Kullanıcı girişi sağlar.

Kullanıcı Bilgileri:

Endpoint: GET /api/user , Kullanıcı profil bilgilerini getirir.

Kategori API'si**Kategori Listeleme:**

Endpoint: GET /api/categories , Tüm kategorileri listeler.

Kategori Oluşturma:

Endpoint: POST /api/categories , Yeni kategori ekler.

Kategori Güncelleme:

Endpoint: PUT /api/categories/{id} , Mevcut kategori günceller.

Kategori Silme:

Endpoint: DELETE /api/categories/{id} , Belirtilen kategori siler.

Ürün API'si**Ürün Listeleme:**

Endpoint: GET /api/products , Tüm ürünleri listeler.

Ürün Ekleme:

Endpoint: POST /api/products , Yeni ürün ekler.

Ürün Güncelleme:

Endpoint: PUT /api/products/{id} , Mevcut ürünü günceller.

Ürün Silme:

Endpoint: DELETE /api/products/{id} , Belirtilen ürünü siler.

Sepet API'si**Sepete Ürün Ekleme:**

Endpoint: POST /api/cart/add , Sepete ürün ekler.

Sepetten Ürün Çıkarma:

Endpoint: DELETE /api/cart/remove/{productId} , Sepetten ürünü çıkarır.

Sepetteki Ürünü Güncelleme:

Endpoint: PUT /api/cart/update/{productId} , Sepetteki ürün miktarını günceller.

Sepeti Boşaltma:

Endpoint: DELETE /api/cart/empty , Sepeti boşaltır.

Sipariş API'si**Sipariş Oluşturma:**

Endpoint: POST /api/orders , Yeni sipariş oluşturur.

Faturalandırma ve Yazdırma:

Endpoint: GET /api/orders/{id}/invoice , Sipariş için fatura oluşturur.

Veri Görselleştirme API'si**İstatistiksel Analiz:**

Endpoint: GET /api/stats , Sistem verileri üzerinde istatistiksel analiz yapar.

Frontend (Ön Yüz) İçeriği

POS uygulamasının ön yüzü, kullanıcı deneyimini artırmak ve işlemleri kolaylaştırmak amacıyla çeşitli bileşenler ve sayfalardan oluşur. React kullanılarak geliştirilecek olan ön yüz, aşağıdaki bölümleri içermektedir:

Giriş ve Kayıt Sayfaları

Giriş ve kayıt sayfaları, kullanıcıların sisteme kaydolmasını ve mevcut hesaplarına giriş yapmasını sağlar. E-posta ve şifre doğrulaması ile güvenli erişim sağlanır.

Ana Sayfa

Ana sayfa, kullanıcıya genel bir kontrol paneli sunar. Ürünler burada görünmeye başlar.

Kategori Yönetimi Sayfası

Kategori yönetimi sayfası, kullanıcıların yeni kategoriler ekleyip mevcut olanları güncelleyebileceği veya silebileceği bir arayüz sunar.

Ürün Yönetimi Sayfası

Ürün yönetimi sayfası, ürünlerin eklenmesi, güncellenmesi ve listelenmesi için bir arayüz sağlar. Ürünler kategorilere göre organize edilir.

Sepet Sayfası

Sepet sayfası, kullanıcıların sepete ekledikleri ürünleri yönetmesine olanak tanır. Ürün miktarları güncellenebilir ve sepet gerektiğinde tamamen boşaltılabilir.

Sipariş ve Fatura Sayfası

Sipariş ve fatura sayfası, kullanıcıların sepetlerindeki ürünlerle sipariş oluşturmaya ve bu siparişlerin faturalandırılmasına olanak tanır. Faturalar PDF olarak indirilebilir ve yazdırılabilir.

Veri Görselleştirme Sayfası

Veri görselleştirme sayfası, satış ve stok durumu gibi verilerin grafiksel ve tablo şeklinde sunulmasını sağlar. Ant Design Charts kullanılarak veriler etkili bir şekilde görselleştirilir.

Backend (Arka Taraf) İçeriği

Kullanıcı Yönetimi

Kullanıcı yönetimi, kullanıcı kayıt ve giriş işlemlerini, kimlik doğrulama ve yetkilendirme süreçlerini kapsar. JWT kullanılarak güvenli oturumlar sağlanır.

Kategori ve Ürün Yönetimi

Kategori ve ürün yönetimi, CRUD işlemleri için API uç noktalarını içerir. Ürünler ve kategoriler için veritabanı işlemleri gerçekleştirilir.

Sepet ve Sipariş Yönetimi

Sepet ve sipariş yönetimi, kullanıcıların sepetlerini yönetmesine ve sipariş oluşturmaya olanak tanır. Siparişler için fatura oluşturma işlemleri de dahildir.

Veritabanı Yönetimi

MongoDB, uygulamanın veri depolama ve erişim ihtiyaçlarını karşılar. Veritabanı işlemleri için Go ile yazılmış API'lar kullanılır.

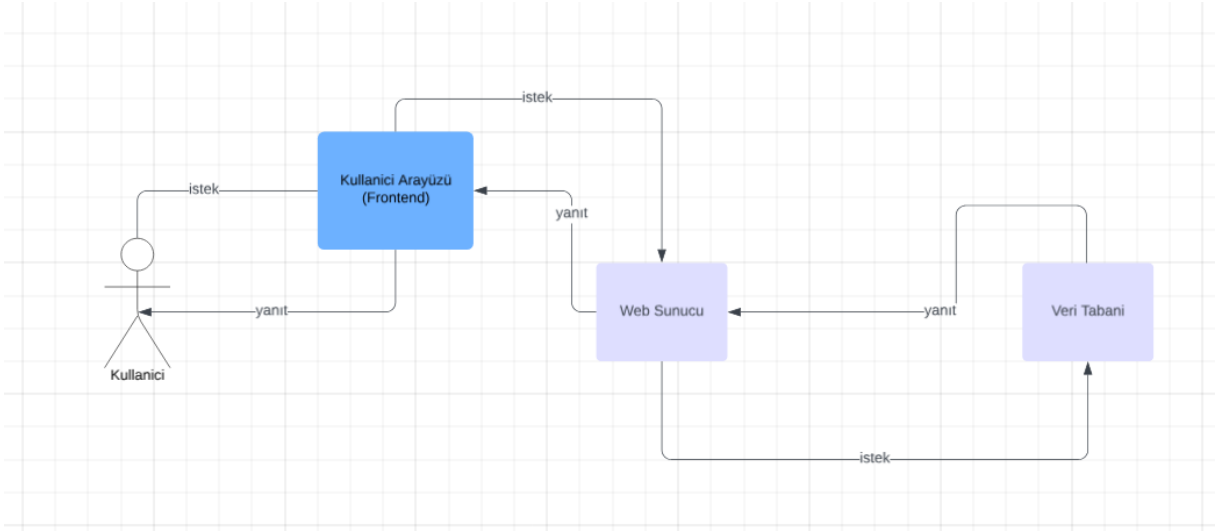
Veri Görselleştirme

Veri görselleştirme, API üzerinden veritabanından elde edilen verilerin analiz edilmesi ve görselleştirilmesi sürecini içerir. İstatistikler, grafik ve tablolar aracılığıyla sunulur.

Ana Mimari ve Akış Diyagramları

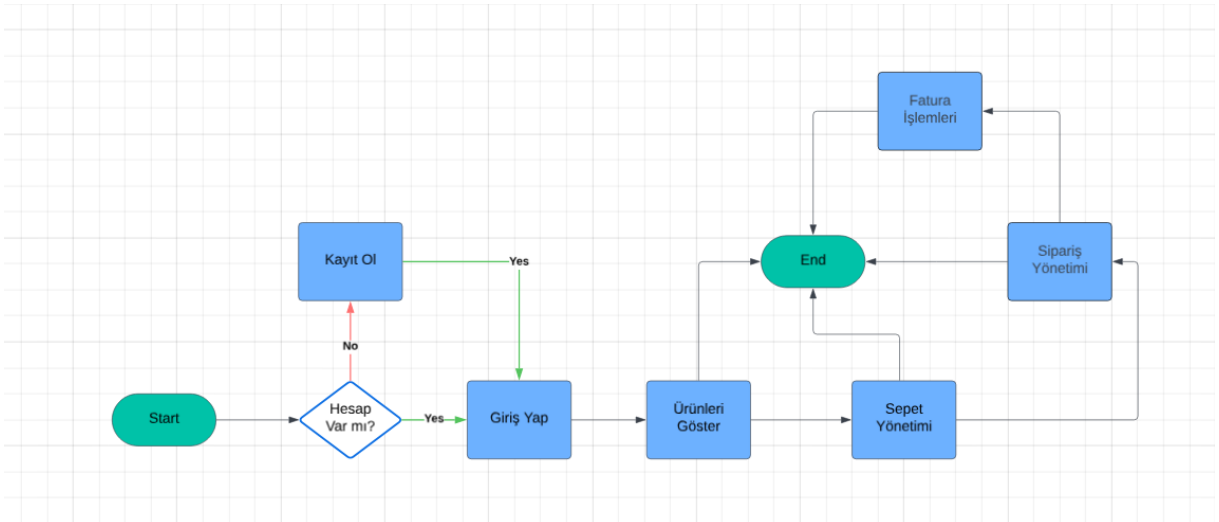
Ana Mimari

Uygulamanın ana mimarisi, frontend (React) ve backend (Go) arasında veri alışverişini sağlayan RESTful API'lar üzerinden çalışır. MongoDB, veritabanı olarak kullanılır ve Docker, uygulamanın taşınabilirliğini artırmak için kullanılır.



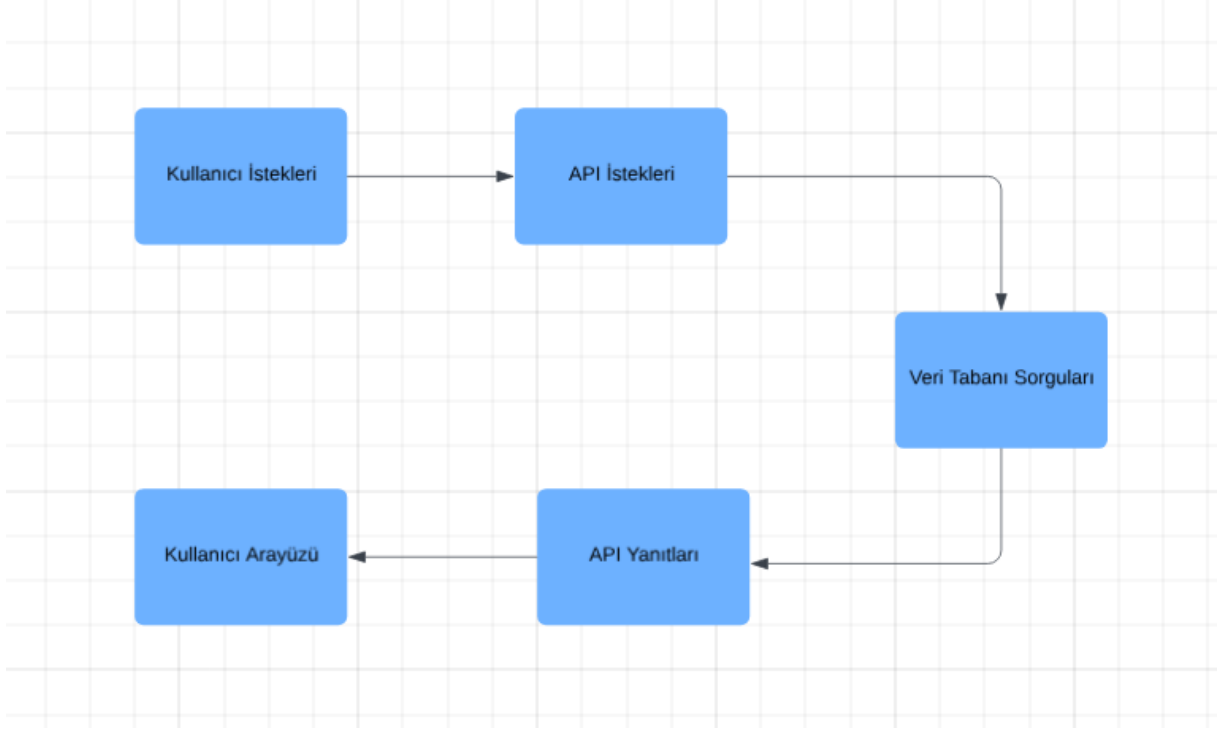
Kullanıcı Akış Diyagramları

Kullanıcı akış diyagramları, kullanıcıların uygulama içerisindeki etkileşimlerini ve süreçlerini gösterir. Örneğin, kullanıcı kaydı, giriş yapma, ürün arama, sepet yönetimi ve sipariş oluşturma gibi işlemleri içerir.



Veri Akış Diyagramları

Veri akış diyagramları, uygulama içinde verilerin nasıl işlendiğini ve aktarıldığını gösterir. Veritabanı sorguları, API istek ve yanıtları gibi süreçlerin görselleştirilmesine olanak tanır.



Test Senaryoları

Birim Testleri (Unit Tests)

Birim testleri, uygulamanın en küçük parçalarının (genellikle fonksiyonlar veya sınıflar) bağımsız olarak doğru çalışıp çalışmadığını kontrol etmek için kullanılır.

Kullanıcı Kaydı ve Girişi:

Kayıt ve giriş işlemleri için kullanılan fonksiyonlar, doğru sonuçlar veriyor mu kontrol edilir.

Hatalı girdi durumlarında uygun hata mesajları gösteriliyor mu kontrol edilir.

Profil Yönetimi:

Profil bilgilerini güncelleyen fonksiyonların doğru çalıştığı ve beklenen sonuçları ürettiği kontrol edilir.

Kategori Yönetimi:

Kategori ekleme, güncelleme ve silme işlemleri için fonksiyonların doğru çalıştığı kontrol edilir.

Ürün Yönetimi:

Ürün ekleme, güncelleme ve silme işlemleri için kullanılan fonksiyonların doğruluğu kontrol edilir.

Sepet Yönetimi:

Sepete ürün ekleme, çıkarma ve miktar güncelleme fonksiyonlarının doğru çalıştığı kontrol edilir.

Fatura Oluşturma:

Fatura oluşturma fonksiyonlarının doğru sonuçlar ürettiği ve beklenen formatta çıktılar sağladığı kontrol edilir.

Entegrasyon Testleri (Integration Tests)

Entegrasyon testleri, farklı modüllerin veya bileşenlerin bir arada uyum içinde çalışıp çalışmadığını test etmek için kullanılır.

Kullanıcı Kaydı ve Girişi:

Kayıt ve giriş işlemlerinin API ve veritabanı ile entegrasyonunun doğru çalıştığı kontrol edilir.

Profil Yönetimi:

Profil güncellemelerinin API ve veritabanı ile entegrasyonunun doğru çalıştığı test edilir.

Kategori Yönetimi:

Kategori işlemlerinin API ile veritabanı arasında doğru entegre edildiği kontrol edilir.

Ürün Yönetimi:

Ürün ekleme, güncelleme ve silme işlemlerinin veritabanı ve API ile uyumlu çalıştığı kontrol edilir.

Sepet ve Sipariş Yönetimi:

Sepet işlemlerinin API ve veritabanı ile doğru entegre edildiği kontrol edilir.

Sipariş oluşturma sürecinin tüm bileşenler (sepet, ürün, kullanıcı) ile uyumlu çalıştığı test edilir.

Veri Görselleştirme:

Grafik ve tabloların veritabanı ve API'dan doğru veri çekip çekmediği kontrol edilir.

Özet

Bu analiz dokümanı, React, Go ve MongoDB kullanılarak geliştirilecek olan POS uygulamasının kapsamlı bir özetini sunmaktadır. Projenin temel amacı, kullanıcı dostu, hızlı ve güvenilir bir POS sistemi oluşturmaktır. Aşağıda projenin ana noktaları özetlenmiştir:

Projenin Amacı ve Kapsamı

POS uygulaması, kullanıcıların ürün yönetimi, sipariş işlemleri, faturalandırma ve veri görselleştirme gibi işlemleri kolayca gerçekleştirmesine olanak tanıyacak şekilde tasarlanmıştır. Uygulama, kullanıcıların günlük iş süreçlerini verimli bir şekilde yönetmelerine yardımcı olmayı hedeflemektedir.

Teknoloji Yığını

Proje, React ile modern ve kullanıcı dostu bir arayüz sunarken, Go ile performanslı ve güvenilir bir backend sağlamaktadır. MongoDB veritabanı olarak kullanılacak ve veri depolama, erişim ve yönetimi için kullanılacaktır.

Ana Özellikler

Kullanıcı Yönetimi: Kayıt, giriş, profil yönetimi

Ürün ve Kategori Yönetimi: Ürün ve kategori ekleme, güncelleme, silme

Sepet ve Sipariş Yönetimi: Sepet işlemleri, sipariş oluşturma, fatura yönetimi

Veri Görselleştirme: Satış ve stok verilerinin grafiksel gösterimi

Test Senaryoları

Uygulamanın doğruluğunu ve güvenilirliğini sağlamak için birim testleri, entegrasyon testleri uygulanacaktır.