

Online Yemek Sipariş Sistemi

REST Api - Projesi

Amaç: Java Spring Boot kullanarak, restoranların menülerini yönetebildiği ve kullanıcıların sipariş verebildiği bir REST API geliştmesi amaçlanmaktadır. RestApi, JPA, DTO, Validation, Spring Security, Rol tabanlı uygulama, JWT, PostgreSQL ve Global Exception Handling konuları bu senaryo üzerinde uygulanacaktır.

Temel Gereksinimler:

Aşağıdaki özelliklerin API tarafından desteklenmesi **zorunludur**:

1. Kullanıcı Yönetimi:

- Yeni kullanıcı kaydı (ad, soyad, e-posta, şifre vb.).
- Kullanıcı girişi (e-posta ve şifre ile JWT tabanlı kimlik doğrulama).
- Rol tabanlı yetkilendirme:
 - **Müşteri (Customer):** Sisteme kayıt olabilir, restoranları ve menülerini görüntüleyebilir(Cache), sipariş oluşturabilir ve geçmiş siparişlerini görüntüleyebilir.
 - **Restoran Sahibi (RestaurantOwner):** Sisteme kayıt olabilir, kendi restoran bilgilerini (ad, adres, iletişim bilgileri vb.) yönetebilir, menüsünü (yemek ekleme, güncelleme, silme, listeleme) yönetebilir ve kendisine verilen siparişleri görüntüleyebilir.
 - **Admin:** Tüm kullanıcıları ve restoranları listeleyebilir(cache, sayfalama), kullanıcı ve restoran bilgilerini güncelleyebilir ve kullanıcı rollerini atayabilir.
- Kullanıcı bilgilerini güncelleme (kendi bilgilerini).

2. Restoran Yönetimi:

- Yeni restoran kaydı (restoran sahibi tarafından).
- Restoran bilgilerini güncelleme (sadece restoran sahibi tarafından).
- Restoranları listeleme (isim veya kategoriye göre filtreleme imkanı).

3. Menü Yönetimi:

- Restorana yeni yemek kategorisi ekleme, düzenleme, silme (sadece restoran sahibi tarafından).
- Restorana (kategoriye göre) yeni yemek ekleme (isim, açıklama, fiyat) (sadece restoran sahibi tarafından).
- Yemek bilgilerini güncelleme (sadece restoran sahibi tarafından).
- Yemek silme (sadece restoran sahibi tarafından).
- Restoranın menüsünü listeleme.

4. Sipariş Yönetimi:

- Müşteri tarafından yeni sipariş oluşturma (hangi restorandan, hangi menü öğelerinden kaç adet).
- Sipariş detaylarını görüntüleme (müşteri ve restoran sahibi tarafından).
- Sipariş durumunu güncelleme (örneğin: "Alındı", "Hazırlanıyor", "Kargoda", "Teslim Edildi") (sadece restoran sahibi tarafından).
- Müşterinin geçmiş siparişlerini listeleme.

- Restoran sahibinin kendisine verilen siparişleri listeleme (duruma göre filtreleme imkanı).

Teknolojiler:

Aşağıdaki teknolojilerin kullanılması **zorunludur**:

- Java
- Spring Boot
- Spring Web (RestApi geliştirme için)
- Spring Data JPA (Veritabanı işlemleri için)
- PostgreSQL (Veritabanı)
- Spring Security (Kimlik doğrulama ve yetkilendirme için)
- JSON Web Tokens (JWT) (Oturum yönetimi için)
- Bean Validation API (Veri validasyonu için - @Valid ve ilgili annotation'lar)
- DTO (Data Transfer Object) pattern'i (API katmanı ile entity katmanı arasındaki veri transferi için)
- Global Exception Handling (Uygulamada oluşan hataları merkezi bir şekilde yönetmek ve anlamlı hata yanıtları dönmek için @RestControllerAdvice ve @ExceptionHandler)
- Caching
- Sayfalama - Pageable
- Swagger kullanarak API dökümantasyonu oluşturma.

Teslim Edilmesi Gerekenler:

- Proje kaynak kodu GitHub reposu olarak açıklayıcı ve anlaşılır bir şekilde gönderilmelidir.
- Çalışır durumda bir API.
- API endpoint'lerini ve kullanılan DTO'ları açıklayan bir Swagger dokümanı hazırlanmalıdır.

Değerlendirme Kriterleri:

Projeler aşağıdaki kriterlere göre değerlendirilecektir:

- Temel gereksinimlerin eksiksiz ve doğru bir şekilde implemente edilmesi.
- Kodun kalitesi, okunabilirliği ve düzeni.
- API tasarımının prensiplere uygunluğu (RESTful API standartları).
- Güvenlik önlemlerinin doğru uygulanması (Spring Security ve JWT).
- Veri validasyonunun doğru bir şekilde yapılması.
- Global Exception Handling mekanizmasının etkinliği.
- Veritabanı tasarımının uygunluğu.